BAYESIAN VARIABLE SELECTION IN HIGH DIMENSIONAL GENOMIC STUDIES USING NONLOCAL PRIORS

A Dissertation

by

AMIR NIKOOIENEJAD

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Valen E. Johnson
Committee Members,	Wenyi Wang
	Anirban Bhattacharya
	Natarajan Sivakumar
Head of Department,	Valen E. Johnson

December 2017

Major Subject: Statistics

ABSTRACT

The advent of new genomic technologies has resulted in production of massive data sets. The outcomes in such experiments are often binary vectors or survival times, and the covariates are gene expressions obtained from thousands of genes under study. Analysis of these data, especially gene selection for a specific outcome, requires new statistical and computational methods. In this dissertation, I address this problem and propose one such method that is shown to be advantageous in selecting explanatory variables for prediction of binary responses and survival times. I adopt a Bayesian approach that utilizes a mixture of nonlocal prior densities and point masses on the regression coefficient vectors. The proposed method provides improved performance in identifying true models and reducing estimation and prediction error rates in a number of simulation studies for both binary and survival outcomes.

I also describe a computational algorithm that can be used to implement the methodology in ultrahigh-dimensional settings ($p \gg n$). In particular, for survival response datasets I show that MCMC is not feasible and instead provide a computational algorithm based on a stochastic search algorithm that is scalable and p invariant.

As part of the variable selection methodology, I also propose a novel approach for setting prior hyperparameters by examining the total variation distance between the prior distributions on the regression parameters and the distribution of the maximum likelihood estimator under the null distribution. An R package, BVSNLP, is also introduced in this dissertation as a final product which contains all described methodology here. It performs high dimensional Bayesian variable selection for binary and survival outcome datasets that is expected to have a variety of applications including cancer genomic studies.

Another problem that is addressed in this dissertation is methodology for deriving and

extending Uniformly Most Powerful Bayesian tests (UMPBTs) from exponential family distributions to a larger class of testing contexts. UMPBTs are an objective class of Bayesian hypothesis tests that can be considered the Bayesian counterpart of classical uniformly most powerful tests. However, they have previously been exposed for application in one parameter exponential family models. I introduce sufficient conditions for the existence of UMPBTs and propose a unified approach for their derivation. An important application of my methodology is the extension of UMPBTs to testing whether the noncentrality parameter of a χ^2 distribution is zero.

DEDICATION

To my darling *Borna* and beloved *Arezou*

To my *Mom* and *Dad*, the most caring parents in the world

ACKNOWLEDGMENTS

I would like to express my deepest gratitude towards my advisor and mentor forever, Dr. Valen E. Johnson for his generous support throughout my doctoral studies. He was always there when needed, regardless of the type of my hardship. Ethics, manners and style of research are his main attributes I admire the most and will remember forever. Thank you Val, you have been a great inspiration to me and a perfect role model for my professional career.

I thank all the members of my dissertation committee, especially Dr. Wenyi Wang. She acquainted me with the marvelous world of cancer genomics and accepted me as one of her group members in MD Anderson Cancer Center, in the first year of my Ph.D. studies. Thanks also to Dr. Michael Longnecker for his all time support for the graduate students in the department and giving me the opportunity to teach an undergraduate course for two semesters.

I am obliged to the Texas A&M High Performance Research Computing (HPRC) and their staff for providing a powerful cluster which facilitated the simulation and real data analyses of my dissertation. Without them, it was impossible to accomplish my doctoral research objectives.

Special thanks to the kind, friendly and responsible staff of the statistics department. Sandra, Athena, Deanna, Elaine and Andrea, I appreciate your help in guiding me dealing with other necessary aspects of graduate student life besides academic research.

While studying in College Station, I was fortunate enough to be around very wonderful and kindhearted friends who were like family far from the family. Friends who were supportive through highs and lows and were always there to hangout with on weekends when shaking off stress from an onerous week seemed vital. Mehdi, Somayeh, Mohsen,

Hoda, Zoya and Peyman, without you this journey would not be as invigorating and joyful. Thank you!

I am deeply indebted to my family. First and foremost, my parents and my parents-in-law for their unflinching support from thousands miles away, and then my sisters Nastaran and Yasaman and my brother-in-law, Payam. Their encouragement, in all possible ways, has always bolstered my confidence and helped me advancing through my graduate studies here in the US.

Last but not the least, I want to dedicate this work to my life companion and the love of my life, Arezou. She was the one who saw the potential in me and helped me succeed in what seemed infeasible at the beginning: withdrawing from Ph.D. of electrical engineering after two years and re-applying to the Ph.D. of statistics. That turned out to be one of the best decisions I have made in my life. I would never forget her support and encouragement through hard days of my doctoral studies. My dear, without you none of this would have been possible and you are always my courage in my academic quest.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Valen E. Johnson, Dr. Wenyi Wang and Dr. Anirban Bhattacharya of the Department of statistics and Dr. Natarajan Sivakumar of the Department of mathematics.

Parts of the cancer genomic data analyzed in Chapter 3 and 4 were provided by Dr. Wenyi Wang and her group in the department of Bioinformatics and Computational Biology in University of Texas MD Anderson Cancer Center.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by Graduate Assistant Teaching (GAT) and Graduate Assistant Non-Teaching (GANT) funds from the department of statistics at Texas A&M University, in addition to Graduate Assistant Research (GAR) funding from National Institute of Health grant (R01CA158113) and National Cancer Institute grant (1R01CA174206-01).

NOMENCLATURE

AUC	Area Under Curve
BMA	Bayesian Model Averaging
BVS	Bayesian Variable Selection
CRAN	Comprehensive R Archive Network
GLM	Generalized Linear Models
HPPM	Highest Posterior Probability Model
HFM	Highest Frequency Model
iMOM	Inverse Moment Prior
ISIS	Iterative Sure Independence Screening
MCMC	Monte Carlo Markov Chain
MOM	Moment Prior
MPI	Message Passing Interface
MPM	Median Probability Model
NLP	Nonlocal Prior
pMOM	Product Moment Prior
piMOM	Product Inverse Moment Prior
ROC	Receiver Operating Characteristic
TCGA	The Cancer Genome Atlas
UMPBT	Uniformly Most Powerful Bayesian Test

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Motivation and Background

The emergence of mircoarray data in late 1990's and the advent of high throughput gene sequencing technology in mid 2000's introduced a new era which led to the production of numerous ultrahigh dimensional gene expression datasets. The Cancer Genome Atlas (TCGA) and International Cancer Genome Consortium (ICGC) are generating large amounts of high dimensional genomic data making those data available to the research community. These data are generated from myriad studies performed by independent researchers and include DNA copy-number alterations (CNA), mRNA and microRNA expressions and other types of gene-related explanatory variables. An online portal named cBioPortal (Gao et al., 2013) facilitates accessing these datasets via an intuitive Web interface where researchers and clinicians can do various analysis as well as downloading desired data.

This revolution has prompted a new direction in statistical data analysis as well as biomedical and bioinformatics research. Traditional statistical methods cannot be applied to datasets with small samples and very large numbers of covariates. This topic has inspired various statisticians from both frequentist and Bayesian schools of thought and resulted in development of new methodologies. The goal of variable selection in high dimensional data is to identify small subset of covariates that are associated with an outcome. This, imposes a sparsity assumption on the problem. In the context of cancer genomics, the target is to determine genes that are associated with the response vector, which can be continuous, binary or a survival time. Interested readers can refer to Guyon and Elisseeff (2003) for more discussion on objectives of variable selection and its related problems.

A general model for variable selection may be posed as follows,

$$\mathbb{E}(\mathbf{y}_n) = F(\mathbf{X}\boldsymbol{\beta}), \tag{1.1}$$

where $\mathbf{y}_n$ is the response vector, $\mathbf{X}$ is $n \times p$ design matrix and $\boldsymbol{\beta}$ is $p \times 1$ coefficient vector. In ultrahigh dimensional settings $p \gg n$. Depending on values of response vector $\mathbf{y}_n$, this modeling framework encompasses linear regression, logistic regression and other types of generalized linear models. The sparsity assumption implies that the majority of elements in $\boldsymbol{\beta}$ are zero, and thus the sparse selection problem is basically identifying the non-zero elements in $\boldsymbol{\beta}$.

A number of methods have been proposed to address this problem. These include the LASSO (Tibshirani, 1996), a penalized likelihood method that maximizes a product of the likelihood function and a constraint on the sum of the absolute value of components of the regression coefficient $\boldsymbol{\beta}$. A closely related method called Smoothly Clipped Absolute Deviation (SCAD) (Fan and Li, 2001) uses a non-convex penalty function and has been demonstrated to have certain oracle properties in idealized asymptotic settings. Other penalized likelihood functions include the adaptive LASSO (Zou, 2006) and the Dantzig selector (Candes and Tao, 2007). These methods share asymptotic properties similar to SCAD. Correspondingly, Efron et al. (2004) proposed Least Angle Regression (LARS), a variable selection method which is a less greedy version of forward selection methods.

In ultrahigh dimensions ($p \gg n$), an effective computational technique for implementing the techniques described above is the Iterative Sure Independence Screening (ISIS) procedure (Fan and Lv, 2008), which iteratively performs a correlation screening step to reduce the number of explanatory variables so that penalized likelihood methods can be applied. ISIS has been used in conjunction with several penalized likelihood methods—including adaptive LASSO (Zou, 2006), the Dantzig Selector (Candes and Tao, 2007),

and SCAD (Fan and Li, 2001)—to perform model selection. Elastic net (Zou and Hastie, 2005) can also be included in the list of variable selection methods suited for ultrahigh dimensional datasets.

Besides penalized likelihood methods, Guyon et al. (2002) proposed an algorithm by exploiting support vector machine methods based on recursive feature elimination. Wang et al. (2005) take a combination of machine learning algorithms such as decision trees and with a correlation based feature selector to perform gene selection.

A number of Bayesian methods have also been proposed for variable selection by specifying a prior distribution on $\boldsymbol{\beta}$ vector. Notable among these are the approaches proposed by George and McCulloch (1997), which used a mixture-of-normals approximation to spike-and-slab priors on the regression coefficients. Rossell et al. (2013) and Johnson and Rossell (2012) also exploit the two component mixture prior where nonlocal priors are used for non-zero components of the coefficient vector. Rossell et al. (2013) address the problem of identifying variables with high predictive power. Along similar lines, Shin et al. (2015) utilized nonlocal priors for linear regression and showed under some regularity conditions the model selection procedure is consistent when $\log(p) = O(n^{\alpha})$. Lee et al. (2003) proposed a hierarchical probit model along with MCMC based stochastic search to perform gene selection in high dimensional settings using a latent response variable and Gaussian priors on model coefficients. West et al. (2000) provided a Bayesian approach to this problem employing singular value regression and classes of informative prior distributions to estimate coefficients in high dimensional settings. Liang et al. (2008) studied mixtures of g priors for Bayesian variable selection as an alternative to default g priors to overcome several consistency issues associated with the default g prior densities. Hans (2009) proposed Bayesian LASSO where the prediction of future observations is also discussed via the posterior predictive distribution. For other significant priors for model coefficients for variable selection, one can refer to Bae and Mallick (2004) for Normal, Laplace and

Jefferey's prior, Carvalho et al. (2010) for horseshoe prior and Bhattacharya et al. (2015) for Dirichlet-Laplace prior. Cawley and Talbot (2006) utilized non-informative Jeffery's prior along with an improved algorithm named BLogReg classification to reduce computational cost in logistic regression gene selection problem.

The aforementioned methods considered variable selection problem in either linear regression or generalized linear models. For variable selection models for survival times, many of common penalized likelihood methods originally introduced for linear regression have been extended to survival data as well. These include Tibshirani et al. (1997) where the LASSO penalty is imposed on the coefficients in survival analysis, similar to the linear regression problem. Zhang and Lu (2007) utilized adaptive LASSO methodology for time to event data while Antoniadis et al. (2010) adopted the Dantzig selector for survival outcome. The extension of non-convex penalized likelihood approaches, in particular SCAD, to the Cox proportional hazard model is discussed in Fan and Li (2002). The ISIS approach is also extended for ultrahigh dimensional survival data in Fan et al. (2010) where it is used on Cox proportional hazard models and the SCAD penalty is employed for variable selection.

Some Bayesian approaches have also been proposed to address this problem. Faraggi and Simon (1998) proposed a method based on approximating the posterior distribution of the parameters in the proportional hazard model where they define a Gaussian prior on a vector of coefficients. A loss function is then defined in order to select a parsimonious model. A semi-parametric Bayesian approach is utilized by Ibrahim et al. (1999), where they employ a discrete gamma process for the baseline hazard function and a multivariate Gaussian prior for the coefficient vector. Sha et al. (2006) considered Accelerated Failure Time (AFT) models along with data augmentation to impute censored times. A mixture prior in a similar fashion to George and McCulloch (1997) is exploited for sparse selection procedure.

Due to the huge computational load of Bayesian data analysis imposed by Monte Carlo Markov Chain (MCMC) procedure, in particular for high dimensional survival data, the frequentist approaches outnumber their Bayesian counterparts in real genomic applications. Consequently, developing a fairly fast Bayesian variable selection method for high dimensional datasets that can outperform dominant frequentist algorithms seems compelling.

1.2 Main Contribution to the Problem

As described in the previous section, it is encouraging to develop a fast and precise Bayesian variable selector to be applied to various datasets. In this dissertation, I propose a Bayesian hierarchical model where I use a mixture of point mass probabilities and nonlocal priors for vectors of coefficients. The targets of my methodology are cases when the response vector is binary or a survival time. For the former, a logistic regression model is used while for the latter, I utilize Cox proportional hazard models (Cox, 1972). A key feature of my methodology is the automatic selection of hyperparameters of nonlocal priors. In addition, I adopt the stochastic search algorithm with screening introduced by Shin et al. (2015) for survival data in order to make the algorithm scalable and hence invariant to the number of covariates, p.

By testing my algorithm in various simulation datasets under different settings of sample size, number of covariates and correlation matrices, I found the output results were more precise (less false positives in selected variables) and had smaller coefficient estimation error rates in comparison with existing methods. I also applied my algorithm to different important cancer genomics datasets under both binary and survival time scenarios. Those include the Golub leukemia data (Golub et al., 1999), renal cell carcinoma (Cancer Genome Atlas Research Network, 2013) and the AML leukemia dataset introduced in Papaemmanuil et al. (2016). In all cases, my method picked sparser models with

better predictive accuracy.

Another contribution of this dissertation is the development of an R package named BVSNLP to implement the proposed models and make them accessible to researchers. Within each iteration of the algorithm, various nonlinear optimization procedures, as well as Laplace approximation to approximate marginal probability of the data, are performed. These calculations incur immense computational burden. As a result, I implement the models in C++ in order to speed up the computation. Parallel computing ability is another feature of the BVSNLP package. Coupling algorithm in logistic regression variable selection as well as parallel stochastic search algorithm in survival variable selection are algorithms in the package that are benefited from this feature. These are discussed in detail in the forthcoming chapters.

In addition to variable selection, I also studied Uniformly Most Powerful Bayesian Tests, UMPBTs, to extend the work by Johnson (2013c) to a more general class of sampling distributions by providing a sufficient condition for the existence of such tests, as well as a general approach to derive them. The primary application for this extended work is in testing the non-centrality parameter in χ^2 statistics with arbitrary degrees of freedom being equal to zero. This is largely used in contingency tables, χ^2 tests, likelihood ratio tests or even model selection procedures (Hu and Johnson, 2009).

The following chapters are organized as follows. In Chapter 2 I discuss the preliminaries which include Bayesian hierarchical models, a brief review on nonlocal prior densities, the proposed algorithm for hyperparameter selection and the general scheme of Bayesian model selection procedures. Chapter 3 explains my method for binary response data in detail with simulation and real data results. Chapter 4 extends the methodology to datasets with survival time outcomes. The research reported for binary response vectors in Chapter 3 has been published in *Bioinformatics* (Nikooienejad et al., 2016). The extension to UMPBTs and its existence conditions are discussed in Chapter 5, where some examples

of its application to contingency tables are provided. The aforementioned R package is introduced in Chapter 6 where each of its important functions are investigated in detail. Concluding remarks appear in Chapter 7.

2. BAYESIAN HIERARCHICAL MODELS, NONLOCAL PRIORS AND HYPERPARAMETER SELECTION*

2.1 Introduction

High dimensional variable selection problems were introduced in Chapter 1, and a review of common approaches that have been proposed in the past ten to fifteen years was also provided. The main assumption in such problems is sparsity of the vector of coefficients. Sparsity soluion are achieved by penalizing the likelihood function in frequentist approaches. In Bayesian methods, sparsity is imposed by the prior distribution defined on the coefficients, in conjunction with the prior on the model size. In this dissertation, a hierarchical mixture model is constructed in which $\pi(\mathbf{y}_n \mid \boldsymbol{\beta})$ denotes the likelihood function, $\pi(\boldsymbol{\beta})$ denotes the prior on the coefficients and $\pi(\mathbf{k})$ denotes the probability of model $\mathbf{k}$, which depends only on model size. The choices for $\pi(\boldsymbol{\beta})$, $p(\mathbf{k})$ and the numerical procedure that computes the posterior probabilities are the main characteristics of any Bayesian approach to perform variable selection.

A list of notable sparsity priors proposed for $\boldsymbol{\beta}$ can be categorized into the following. Discrete mixture priors (Johnson and Rossell, 2012; George and McCulloch, 1997), Student t distributions (Tipping, 2001), horseshoe priors (Carvalho et al., 2010), Normal/Jefferey's priors (Bae and Mallick, 2004), Normal/exponential-gamma priors (Griffin et al., 2010), and double exponential densities (West, 1987; Park and Casella, 2008; Pericchi and Smith, 1992; Hans, 2009). For more details consult Polson and Scott (2010).

In this dissertation I use the discrete mixture prior structure that defines a point mass at zero for zero coefficients and a nonlocal continuous distribution for non-zero coefficients.

*Part of this chapter is reprinted with permission of Oxford University Press, from the article: Nikooienejad, A., W. Wang, and V. E. Johnson (2016). Bayesian variable selection for binary outcomes in high-dimensional genomic studies using non-local priors. *Bioinformatics 32*(9), 1338-1345.

In particular, I extend the methodology discussed in Johnson and Rossell (2012) for generalized linear models and Cox proportional hazard models to perform variable selection in high dimensional datasets. The fundamental characteristic of this method is the use of nonlocal priors (Johnson and Rossell, 2010).

In contrast to local priors, nonlocal priors are density functions that are equal to zero at the null value of the random variable. The variable selection problem can be converted to a series of hypothesis tests for coefficients being equal to 0, the null value. As discussed in Johnson and Rossell (2010), where local priors are used, the accumulation of evidence in favor of the true null is not at the same rate as that when the alternative is true. Accordingly, for discrete mixture models, the choice of a prior distribution with that characteristics is expected to improve the overall variable selection outcome. For instance in the context of linear regression, Shin et al. (2015) showed that under certain regularity conditions the selection procedure with nonlocal priors is consistent even when the number of covariates p increases sub-exponentially with the sample size n. Using local prior models leads to the assignment of probability 0 to the true model.

This chapter is organized as follows. In Section 2.2 I review two of the common nonlocal priors used in variable selection. Section 2.3 describes the hierarchical model used in Bayesian variable selection. In Section 2.4 I propose a data-specific algorithm for automatic selection of hyperparameters for nonlocal priors and Section 2.5 concludes the chapter.

2.2 Moment and Inverse Moment Nonlocal Priors

The two nonlocal priors proposed in Johnson and Rossell (2010) are the moment prior (MOM) and the inverse moment prior (iMOM) prior densities. A base distribution is needed to construct moment priors. The choice of this base prior depends on the tail behavior of the parameter under study. One common choice for the base prior is the

standard normal distribution. On the other hand, inverse moment priors have functional forms that are related to the inverse gamma density function. In particular, their behavior near the null value is similar to the behavior of inverse gamma distributions near 0.

The choice of multivariate MOM or iMOM priors for the vector of coefficients seems a natural choice. However, as shown in Johnson and Rossell (2010), the multivariate form of those priors are 0 only when all of the components of the vector are zero. This property does not provide a sufficient penalty for models with coefficient estimates near zero. As a result, the product version of such priors, named pMOM and piMOM, are preferred since hey are zero whenever *any* of the components of the regression vector are equal to zero, and get very small when most of the parameter estimates are close to zero.

The pMOM prior with a normal base density for a vector of size k is defined as

$$\pi(\boldsymbol{\beta} \,|\, \tau, \sigma^2, r) = d_k (2\pi)^{-k/2} (\tau\sigma^2)^{-rk-k/2} |\mathbf{A}_k|^{1/2} \times \exp\Big[-\frac{1}{2\tau\sigma^2} \boldsymbol{\beta}' \mathbf{A}_k \boldsymbol{\beta} \Big] \prod_{i=1}^{k} \boldsymbol{\beta}_i^{2r}, \tag{2.1}$$

where $\tau > 0$ and r is a positive integer called the order of the density. $\mathbf{A}_k$ is a $k \times k$ nonsingular scale matrix. The normalizing constant d_k is independent of σ^2 and τ. The parameter σ^2 is the variance of the base Gaussian density and is usually assumed to be 1.

The piMOM density for a vector of size k, a product of iMOM density functions, is defined as

$$\pi(\boldsymbol{\beta}_{\mathbf{k}} | \tau, r) = \frac{\tau^{rk/2}}{\Gamma(r/2)^k} \prod_{i=1}^{k} |\beta_i|^{-(r+1)} \exp\Big(-\frac{\tau}{\beta_i^2} \Big). \tag{2.2}$$

Here, τ is a scale parameter controlling the dispersion of the prior around zero and r acts similar to the shape parameter in the inverse Gamma distribution and is responsible for the tail behavior of the distribution. As a result, the roles of τ and r are critical in the overall

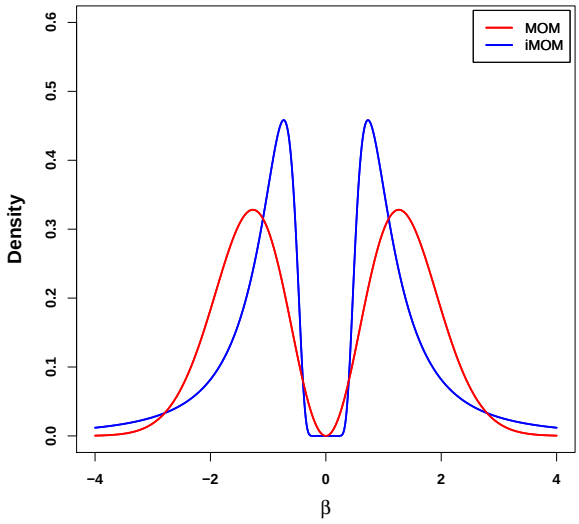


Figure 2.1: pMOM prior with $r = 1$ and $\tau = 0.8$ and piMOM prior with $r = 2$ and $\tau = 0.8$.

performance of the variable selection procedure.

Figure 2.1 depicts an example of both pMOM and piMOM distributions with $k = 1$. For the plots in this figure, the pMOM parameters are $r = 1$ and $\tau = 0.8$. The piMOM has parameters $r = 2$ and $\tau = 0.8$.

As illustrated in Figure 2.1, pMOM's tails are converging to zero at an exponential rate in the same fashion as the Gaussian distribution, while the piMOM's tails are heavier and are therefore more suitable for capturing larger coefficients. Another important point in comparison of these two densities is their behavior in the vicinity of zero. For the piMOM the region where the prior is close to zero is larger that for the pMOM. This region is controlled by the τ parameter in the piMOM density. The larger τ, the wider that region becomes. As mentioned before, the value of τ is crucial in the selection procedure to penalize covariates with very small coefficient estimates and thus reduce the false positive rate. As a result, the piMOM prior is preferred for the applications discussed here.

An important feature of the piMOM nonlocal prior, as highlighted in Johnson and Rossell (2012), is that these priors do not necessarily impose significant penalties on non-sparse models provided that the estimated coefficients in the non-sparse models are not too small. That is, large values of regression coefficients are not penalized since the value of the exponential kernel in (2.2) tends to 1 as β_i becomes large. This fact lies in stark contrast to most penalized likelihood methods. The penalty provided by this prior is on very small coefficient values which makes it a good choice for model selection. It prevents covariates with negligible coefficient estimates from entering the model.

To see how different values of r and τ change the shape of both pMOM and piMOM priors, refer to *https://amirnik.shinyapps.io/nlpinteractive/*. This webpage provides an interactive graphical interface to visualize the effects of hyperparameters on the two aforementioned nonlocal densities. Each plotted graph can be downloaded for future purposes as well.

2.3 Hierarchical Bayesian Modeling in Variable Selection

Let the response vector in my analysis be $\mathbf{y}_n$ which has size n, the number of observations. It can be a response vector from a linear regression model, a binary vector from a logistic regression model or survival times in a Cox proportional hazard model. In this dissertation, I consider the last two models. Note that all these models involve a coefficient vector, $\boldsymbol{\beta}$, where only few number of its elements have non-zero values. Assuming one of the nonlocal priors discussed before is used as the prior for the coefficients, the following hierarchical model can be defined.

$$\begin{aligned} \mathbf{y}_n \,|\, \boldsymbol{\beta}_{\mathbf{k}} \sim \pi(\mathbf{y}_n \,|\, \boldsymbol{\beta}_{\mathbf{k}}, \mathbf{X}) : &\quad \text{Likelihood function under model } \mathbf{k}. \\ \boldsymbol{\beta}_{\mathbf{k}} \sim \pi(\boldsymbol{\beta}_{\mathbf{k}}) : &\quad \text{Nonlocal prior on the coefficients.} \\ p(\mathbf{k}) : &\quad \text{The probability of model } \mathbf{k}, \end{aligned} \tag{2.3}$$

where $\mathbf{X}$ denotes the $n \times p$ design matrix and $\boldsymbol{\beta}_{\mathbf{k}}$ is the vector of coefficients under model $\mathbf{k}$.

The selection procedure is based on the posterior probability of each model and the model with the highest posterior probability is selected. The posterior probability of model $\mathbf{j}$ is defined as

$$p(\mathbf{j}|\mathbf{y}_n) = \frac{p(\mathbf{j})m_{\mathbf{j}}(\mathbf{y}_n)}{\sum_{k\in\mathcal{J}} p(\mathbf{k})m_{\mathbf{k}}(\mathbf{y}_n)}, \tag{2.4}$$

where $m_{\mathbf{k}}(\mathbf{y}_n)$ denotes the marginal probability of the response vector under model $\mathbf{k}$. The denominator is the normalizing constant that is canceled out when comparing model posteriors. Based on the proposed hierarchical model, the marginal probability of the observed data can be expressed as

$$m_{\mathbf{k}}(\mathbf{y}_n) = \int \pi(\mathbf{y}_n \,|\, \boldsymbol{\beta}_{\mathbf{k}})\pi(\boldsymbol{\beta}_{\mathbf{k}})d\boldsymbol{\beta}_{\mathbf{k}}. \tag{2.5}$$

Usually, this integral cannot be calculated analytically and must be numerically approximated. A common method to approximate the integral is the Laplace approximation (Tierney and Kadane, 1986). It is an efficient method because it involves no iteration and avoids numerical integration. A brief review on the first order Laplace method to approximate the marginal probability of data is discussed in the following.

2.3.1 Laplace Approximation to Marginal Probabilities

The basic idea for Laplace approximation is to approximate the integral

$$I(t) = \int e^{-th(x)} dx, \tag{2.6}$$

where the goal is to find the value of $I(t)$ as t tends to an asymptotic limit, namely infinity. In this structure, the one dimensional function $h(x)$ is assumed to have a minimum at $\hat{x}$.

In statistical problems the parameter in asymptotic analysis is usually the sample size, n, which replaces t in the original formulation. Doing a Taylor series expansion of the function $h(x)$ at its minimum, $\hat{x}$, and some integral calculations, the integral in (2.6) can be rewritten as

$$I(n) \approx e^{-nh(\hat{x})} \big(\frac{2\pi}{n}\big)^{1/2} \hat{h}_2^{-1} \left(1 - \frac{\hat{h}_4 \hat{h}_2^{-4}}{8n} + \frac{5\hat{h}_3^2 \hat{h}_2^{-6}}{24n}\right). \tag{2.7}$$

Here, $\hat{h}_i$ denotes the i^{th} derivative of $h(x)$ computed at $\hat{x}$. The neglected terms in the expansion above are of the order $O(n^{-2})$. As a result, the first order Laplace approximation with a precision to the order of $O(n^{-1})$ is computed as

$$I(n) \approx e^{-nh(\hat{x})} \big(\frac{2\pi}{n}\big)^{1/2} \hat{h}_2^{-1}. \tag{2.8}$$

It can be shown that in the case of multivariate integrals where the variable of integration $\mathbf{x}$ is d dimensional, the Laplace approximation in (2.8) can be expressed as

$$\int e^{-nh(\mathbf{x})} d\mathbf{x} \approx e^{-nh(\hat{\mathbf{x}})} (2\pi)^{d/2} |\Sigma|^{1/2} n^{-d/2}. \tag{2.9}$$

Here, $h(\mathbf{x})$ is assumed to have a local minimum and $|\Sigma|$ is the determinant of the $d \times d$ matrix Σ, which is equal to the inverse of the Hessian of the function $h(\mathbf{x})$, computed at

its extreme point, $\hat{\mathbf{x}}$. That is

$$\Sigma = H_{\hat{\mathbf{x}}}^{-1} = \left[\frac{\partial^2 h(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} \right]_{\mathbf{x}=\hat{\mathbf{x}}}^{-1}. \tag{2.10}$$

To exploit Laplace approximation to approximate marginal probability of data in (2.5), define the function $g(\boldsymbol{\beta}_{\mathbf{k}})$ as the negative of the log posterior function:

$$g(\boldsymbol{\beta}_{\mathbf{k}}) = -\log\big(\pi(\mathbf{y}_n \,|\, \boldsymbol{\beta}_{\mathbf{k}})\big) - \log\big(\pi(\boldsymbol{\beta}_{\mathbf{k}})\big), \tag{2.11}$$

and let $h(\boldsymbol{\beta}_{\mathbf{k}}) = \frac{1}{n} g(\boldsymbol{\beta}_{\mathbf{k}})$. It is then obvious that $m_{\mathbf{k}}(\mathbf{y}_n) = \int e^{-nh(\boldsymbol{\beta}_{\mathbf{k}})} d\boldsymbol{\beta}_{\mathbf{k}}$. In addition, let G and H denote the Hessian of the functions g and h, respectively. The determinants of those matrices will then have the following relation

$$|G| = n^{-d} |H|. \tag{2.12}$$

By plugging these results into (2.9) and using (2.10), the marginal probability of the response vector under model $\mathbf{k}$ is approximated by

$$m_{\mathbf{k}}(\mathbf{y}_n) = \pi(\mathbf{y}_n \,|\, \hat{\boldsymbol{\beta}}_{\mathbf{k}}) \pi(\hat{\boldsymbol{\beta}}_{\mathbf{k}}) (2\pi)^{d/2} |G_{\hat{\boldsymbol{\beta}}_{\mathbf{k}}}|^{-1/2}. \tag{2.13}$$

Here, $|G_{\hat{\boldsymbol{\beta}}_{\mathbf{k}}}|$ is the determinant of the Hessian of function $g(\boldsymbol{\beta}_{\mathbf{k}})$ computed at $\hat{\boldsymbol{\beta}}_{\mathbf{k}}$. Note that the approximation formula does not directly depend on the sample size n.

The $\hat{\boldsymbol{\beta}}_{\mathbf{k}}$ is the maximum a posteriori (MAP) estimate of the model which minimizes the function $g(\boldsymbol{\beta}_{\mathbf{k}})$ as defined in (2.11). Moreover, if such a point exists for $h(\boldsymbol{\beta}_{\mathbf{k}})$, the Hessian is positive definite and consequently all of its eigenvalues are positive. As a result, the determinant of Σ will be positive.

To minimize $g(\boldsymbol{\beta}_{\mathbf{k}})$, different non-linear optimization algorithms can be used. For a

review of such methods available in R, refer to Mullen (2014). The main algorithm I use is limited memory version of the Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS) method for large scale optimization (Liu and Nocedal, 1989) that is implemented as a C++ class (Qiu et al., 2016).

2.3.2 Prior on Model Space

Inclusion of covariates in model $\mathbf{k}$ can be modeled as an exchangeable Bernoulli trials. Let $\boldsymbol{\gamma}_{\mathbf{k}} = \{\gamma_1, \cdots, \gamma_p\}$ be the binary vector showing which covariates are included in model $\mathbf{k}$. The size of the model is the number of nonzero elements in $\boldsymbol{\gamma}_{\mathbf{k}}$. These nonzero indices represent the index of nonzero elements in the coefficient vector, $\boldsymbol{\beta}$. Assume the model size is k and the success probability for the Bernoulli trial is $p(\gamma_i = 1) = \theta$ for every $1 \leq i \leq p$. As discussed in Scott et al. (2010), no fixed value for θ independent of p adjusts for multiplicity. As a result, it is necessary to define a prior on θ. The resulting marginal probability for model $\mathbf{k}$ in a fully Bayesian approach is then

$$p(\mathbf{k}) = \int \theta^k (1-\theta)^{p-k} \pi(\theta) d\theta. \tag{2.14}$$

A common choice for $\pi(\theta)$ is the beta distribution, $\theta \sim \text{Beta}(a, b)$, where in the special case of $a = b = 1$, $\pi(\theta)$ is a uniform distribution. The marginal probability for model $\mathbf{k}$ derived from (2.14) is then equal to

$$p(\mathbf{k}) = \frac{B(a+k, b+p-k)}{B(a,b)}, \tag{2.15}$$

where $B(.)$ is the Beta function. In my analysis I chose $a = p$ and $b = p - a$. With this choice of a and b, the mean and variance of the selected model size, k, is

$$\mathbb{E}(k) = a, \quad \text{Var}(k) = a - \frac{a^2}{p} \approx a. \tag{2.16}$$

Using (2.4) and the Laplace approximation (2.13), the selection procedure is performed either by Monte Carlo Markov Chain (MCMC) or other algorithms to find a model with the highest posterior probability. The algorithm used depends on the structure of the problem and the computational cost. For instance, MCMC is used for logistic regression. However, MCMC is too costly for the Cox proportional hazard model. I discuss the details of the estimation procedure in the later chapters.

2.4 Hyperparameter Selection

A critical aspect of implementing my model is the choice of the hyperparameters r and τ. As mentioned previously, the value of r determines the tail behavior of the piMOM prior. The value of τ plays a role similar to the tuning parameter in penalized likelihood methods, where its value largely determines the minimum value of a component of $\boldsymbol{\beta}_{\mathbf{k}}$ that will be selected into a high posterior probability model.

To pick an appropriate, application-specific value for τ, I adopt a strategy in which I compare the null distribution of the maximum likelihood estimator for $\boldsymbol{\beta}_{\mathbf{k}}$ (i.e., when all components of $\boldsymbol{\beta}_{\mathbf{k}}$ are 0), obtained from a randomly selected design matrix $\mathbf{X}_{\mathbf{k}}$, to the prior density on $\boldsymbol{\beta}_{\mathbf{k}}$ under the alternative assumption that the components are non-zero. By choosing τ to be just large enough so that the intersection of these two densities falls below a specified threshold, I am able to approximately bound the probability of false positives in the model, while at the same time maintaining sensitivity to regression coefficients that fall outside of the distribution of MLEs that estimate 0. In principle, I can employ this strategy to obtain a distinct value of τ for each visited sub-model $\mathbf{k}$, but I was unable to do so in the applications discussed in this dissertation because of the computational expense this procedure would impose. Instead, I mixed over models to obtain a single value of τ.

Numerically, my strategy is implemented as follows. I begin by sampling a model from the prior on the model space. That is, I randomly sample k columns of $\mathbf{X}$ where k

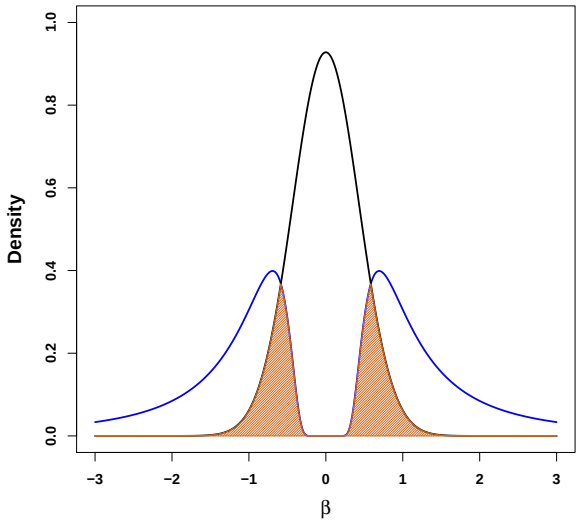


Figure 2.2: Example of overlap between a piMOM prior and approximate normal distribution of null MLE coefficients

is determined by a draw from the prior on the model space. For binary outcome datasets, a Bernoulli vector of length n with success probability $\hat{\pi}$ is generated, where $\hat{\pi}$ is the proportion of successes in the observed data. For survival datasets, survival times have to be estimated. To do this, I use the method discussed in section 4.2.2. Using estimated responses from the null model, the MLE is estimated. This process is repeated N times to obtain a normal density approximation to the marginal density of the maximum likelihood estimates under the condition that all true regression coefficients (except for the intercept in linear or logistic regression) are 0. Typically, $N = O(10^4)$.

Next, piMOM priors corresponding to different values of τ are compared to the null distribution of the MLE. Based on these comparisons, I numerically determine the value of τ so that the overlap of these densities falls below a threshold of $p^{-1/2}$. This overlap value is chosen heuristically in a way that suggests the number of false positives will decrease to 0 as p and n become large. Other thresholds of the form $p^{-\alpha}$ might also be considered,

but I have found that $\alpha = 1/2$ provides good performance in a wide range of simulation studies and in real data examples. Further justification for this threshold is provided in section 2.4.1. Figure 2.2 illustrates the overlap between piMOM prior with $r = 1.5$ and $\tau = 0.6$ and the approximate normal distribution for null MLE values with $\sigma = 0.43$.

Since r controls the tail behavior of the piMOM, I can impose a constraint on the maximum values of estimated coefficients to find an appropriate value for r. For the specific applications discussed in this dissertation, namely logistic regression and the Cox proportional hazard models, it is not sensible to have an absolute value of a coefficient that is more than 10. Therefore I can pick the r value so that $|\beta|$ falls in the interval $(-10, 10)$ with 95% probability. A numerical strategy for finding this hyperparameter vector is outlined in Algorithm 1 below.

Algorithm 1 Choosing Appropriate r and τ for piMOM

procedure RTAUSELECT($\mathbf{X}, n, p$)
  $\mathbf{y_n} \leftarrow$ Sample from the NULL model
  for (i in 1:N) **do**
    ksize $\leftarrow$ Sample from prior on model space in (3.4)
    $\mathbf{X_k} \leftarrow$ Randomly choose *ksize* columns from $\mathbf{X}$
    $\boldsymbol{\beta_i} \leftarrow \text{MLE}(\mathbf{y_n}, \mathbf{X_k})$
    $\boldsymbol{\beta} \leftarrow [\boldsymbol{\beta} \quad \boldsymbol{\beta_i}]$
  $f \leftarrow$ Normal density approximation to density of $\boldsymbol{\beta}$
  ov $\leftarrow$ Overlap area between f and iMOM(τ, r)
  tp $\leftarrow$ Area under iMOM(τ, r) outside the interval $(-10, 10)$
  $[r^*, \tau^*] \leftarrow \underset{r,\tau}{\text{argmin}}(|ov - \frac{1}{\sqrt{p}}| + |tp - 0.05|)$
  return $[r^*, \tau^*]$

Notice that this procedure for choosing the hyperparameters depends on the prior on the model space. This implies that τ will tend to be larger in larger models, because it is more likely that the sampled columns of $\mathbf{X}$ will exhibit high collinearity in large

models. In addition, I recommend this algorithm for cases where $p \gg n$, for instance with $p > 2n$. Otherwise, a very large n can make the distribution of the null MLE very narrow, preventing the desired overlap from being achieved for reasonable values of τ.

2.4.1 Justification For $1/\sqrt{p}$ Overlap

My rationale for setting the overlap between the sampling distribution of the MLE and the prior density to be $p^{-1/2}$ can be explained as follows. For simplicity, I motivate my criterion in the context of a scalar-valued parameter θ. Let $p(\theta)$ denote the prior density for θ under a nonlocal prior defining the alternative hypothesis, H_1, and let $f(\theta) = \prod_{i=1}^{n} f_i(x_i|\theta)$ denote the likelihood function, let $i(\hat{\theta})$ denote the observed information evaluated at the MLE $\hat{\theta}$, i.e.,

$$i(\hat{\theta}) = - \left. \frac{\partial^2 \log f(\theta)}{\partial \theta} \right|_{\theta=\hat{\theta}}. \tag{2.17}$$

Under the null hypothesis, $\theta = 0$ and therefore under the null model the marginal density of the data is simply $m_0 = f(0)$. The marginal likelihood function under the alternative hypothesis can be approximated using Laplace approximation method as

$$m_1(\hat{\theta}) \approx \sqrt{\frac{2\pi}{i(\hat{\theta})}} f(\hat{\theta}) p(\hat{\theta}).$$

In large samples when the null hypothesis is true,

$$f(\hat{\theta}) \approx f(0) e^{\eta(\hat{\theta})/2}, \tag{2.18}$$

where η is a chi-squared random variable, which is bounded in probability. Also, for large

n, the observed information $i(\hat{\theta})$ converges to Fisher's information, $I(0)$. Define w to be

$$w = \sqrt{\frac{2\pi}{I(0)}}. \tag{2.19}$$

Now let $g(\hat{\theta})$ denote the sampling distribution of the maximum likelihood estimate under the null hypothesis. I assume that this sampling density is approximately normally distributed around 0 and let $\pm x$ denote the point at which the sampling density of the MLE and the nonlocal prior densities overlap. Under my constraint on the overlap between densities, the expected value of m_1 satisfies

$$\begin{aligned} E_0[m_1(\hat{\theta})]/w &\approx \int_{|\hat{\theta}|<x} f(0)e^{\eta(\hat{\theta})/2}p(\hat{\theta})g(\hat{\theta})d\hat{\theta} + \int_{|\hat{\theta}|>x} f(0)e^{\eta(\hat{\theta})/2}p(\hat{\theta})g(\hat{\theta})d\hat{\theta} \\ &\leq \max[g(\hat{\theta})] \int_{|\hat{\theta}|<x} f(0)e^{\eta(\hat{\theta})/2}p(\hat{\theta}) + \max[p(\hat{\theta})] \int_{|\hat{\theta}|>x} f(0)e^{\eta(\hat{\theta})/2}g(\hat{\theta})d\hat{\theta} \\ &\leq \max[g(\hat{\theta}), p(\hat{\theta})] \left[\int_{|\hat{\theta}|<x} f(0)e^{\eta(\hat{\theta})/2}p(\hat{\theta}) + \int_{|\hat{\theta}|>x} f(0)e^{\eta(\hat{\theta})/2}g(\hat{\theta})d\hat{\theta} \right] \\ &\approx \max[g(\hat{\theta}), p(\hat{\theta})] f(0)e^{\eta'/2}\frac{1}{\sqrt{p}} \end{aligned} \tag{2.20}$$

for some random variable η' that is bounded in probability. The Bayes factor in favor of the larger model is thus

$$BF_{10} < w \max[g(\hat{\theta}), p(\hat{\theta})] \exp(\eta'/2)\frac{1}{\sqrt{p}}. \tag{2.21}$$

For large n, the second term on the right hand side of the inequality is determined by the sampling distribution of the MLE and is $O_p(n^{1/2})$, while w is $O(n^{-1/2})$. Thus, the average Bayes factor is $O_p(p^{-1/2})$, and combined with the beta-binomial prior on the model space (which imposes a penalty that is $O(1/p)$ on new variables), this suggests that the number

of false positives under the null model of no effects will decrease to 0 as p increases.

2.5 Discussion

In this chapter I discussed the hierarchical Bayesian model used in my algorithm to perform high dimensional variable selection. A brief review of the Laplace approximation was also provided in order to compute the marginal probability of the data.

The main idea of my method is the use of nonlocal priors, in particular the piMOM density, for nonzero coefficients. The piMOM density has better behavior around the origin than pMOM density, as well as heavier tails than pMOM, making it suitable for sparse variable selection. An automatic approach for selecting hyperparameters of the piMOM prior was also discussed.

This was a general description of the methodology. The specific details regarding the selection procedure in binary or survival response datasets are provided in the following chapters.

3. HIGH DIMENSIONAL BAYESIAN VARIABLE SELECTION FOR BINARY RESPONSE DATA*

3.1 Introduction

Recent developments in bioinformatics and cancer genomics have made it possible to measure thousands of genomic variables that might be associated with the manifestation of cancer. The availability of such data has resulted in a pressing need for the development of statistical methods to use these data to identify variables that are associated with binary outcomes (e.g., cancer or control, survival or death). The topic of this chapter is a statistical model for identifying, from a large number p of potential feature vectors, a sparse subset that are useful in predicting a binary outcome vector. Throughout this chapter, I assume that the binary vector of interest is denoted by $\mathbf{y}$, and that the matrix of potential explanatory variables is denoted by $\mathbf{X}$. Along the same lines of (1.1), letting $\mathbf{X_k}$ denote the submatrix of $\mathbf{X}$ containing the "true" predictors, I assume that

$$\boldsymbol{\pi} = F(\mathbf{X_k}\boldsymbol{\beta_k}), \tag{3.1}$$

where F denotes a known binary link function (assumed to be the logistic distribution in what follows), and $\boldsymbol{\pi}$ is the n vector of success probabilities for $\mathbf{y}$. The regression coefficient $\boldsymbol{\beta_k}$ represents the non-zero regression effect for each column of $\mathbf{X_k}$ in predicting $\boldsymbol{\pi}$. The primary statistical challenge addressed in this chapter is the selection of the submatrix $\mathbf{X_k}$ to be used for the prediction of $\boldsymbol{\pi}$.

Different approaches have been proposed to tackle this problem for ultrahigh dimen-

*Part of this chapter is reprinted with permission of Oxford University Press, from the article: Nikooienejad, A., W. Wang, and V. E. Johnson (2016). Bayesian variable selection for binary outcomes in high-dimensional genomic studies using non-local priors. *Bioinformatics 32*(9), 1338-1345.

sional datasets ($p \gg n$) in both Bayesian and frequentist paradigms. Penalized likelihood based methods include the Iterative Sure Independent Screening (ISIS) procedure (Fan and Lv, 2008) which can be applied to different penalty functions such as SCAD (Fan and Li, 2001), LASSO (Tibshirani, 1996) and Dantzig selector (Candes and Tao, 2007). Among Bayesian approaches, Lee et al. (2003) exploit hierarchical probit model with a Gaussian prior for coefficients. West et al. (2000) provided a Bayesian approach to this problem by using informative prior distributions. Liang et al. (2008) studied a mixture of g priors and Bae and Mallick (2004) considered Gaussian, Jeffery's and Laplace priors. George and McCulloch (1997) and Rossell et al. (2013) can be listed among methods that utilized discrete mixture priors.

Except for Rossell et al. (2013), each of the Bayesian methods described above impose local prior densities on regression coefficients in the true model. That is, the prior density on the regression coefficients has a positive prior density function at 0 (and in most cases has its mode at 0), which from a Bayesian perspective makes it more difficult to distinguish between models that include regression coefficients that are close to 0 and those that do not. Johnson and Rossell (2012) proposed two new classes of nonlocal prior densities to ameliorate this problem. In the model selection context, nonlocal prior densities are 0 when a regression coefficient in the model is 0. This makes it easier to distinguish between coefficients that do not have an impact on the prediction of $\mathbf{y}$ from those that do. Johnson and Rossell (2012) used a Markov Chain Monte Carlo (MCMC) algorithm to sample from the posterior distribution on the model space; the convergence properties of this algorithm were studied in Johnson (2013a).

The primary goal of this chapter is to extend the methodology proposed in Johnson and Rossell (2012) for application to binary outcomes and to compare the performance of this algorithm to leading penalized likelihood methods. In addition, I describe a default procedure for setting the hyperparameters (i.e., tuning parameters) in the nonlocal priors,

and I examine a numerical strategy for identifying the highest posterior probability model (HPPM).

The remainder of this chapter is structured as follows. In Section 3.2 I describe the MOMLogit variable selection model for binary regression. As part of this description, I propose a default method for setting the hyperparameter values in the nonlocal priors imposed on the regression coefficients, as well as a numerical strategy for estimating the HPPM. The approach for setting the hyperparameter values is new, and is based on comparing the distribution of the maximum likelihood estimates (MLEs) of randomly selected null models to the nonlocal prior distributions on the regression coefficients. Because the null distribution of the MLEs are centered on 0 and the nonlocal priors decrease to 0 at 0, I show that it is possible to choose model hyperparameters so that the total variation distance between these distributions exceeds a given threshold. Section 3 provides a brief description of a computational procedure designed to identify the HPPM. Section 3.4 presents a simulation study to compare the MOMLogit procedure and ISIS-SCAD algorithm in ultrahigh dimensional settings. My method is then applied to detect genes that are associated with cancer in that section. Finally Section 3.5 concludes with a discussion of the advantages and disadvantages of the MOMLogit variable selection procedure.

3.2 Methods

Let $\mathbf{y}_n = (y_1, \ldots, y_n)^T$ denote a vector of independent binary observations, $\mathbf{X}_n$ an $n \times p$ matrix of real numbers, $\boldsymbol{\beta}$ a $p \times 1$ regression vector, and $\mathbf{x}_i$ the i^{th} row of $\mathbf{X}_n$. I denote a model by $\mathbf{k} = \{k_1, \ldots, k_j\}$ where $(1 \leq k_1 < \cdots < k_j \leq p)$ and it is assumed that $\beta_{k_1} \neq 0, \ldots, \beta_{k_j} \neq 0$ and all other elements of $\boldsymbol{\beta}$ are 0. The design matrix corresponding to model $\mathbf{k}$ is denoted by $\mathbf{X_k}$ and is defined to have cardinality k. I assume that the columns of $\mathbf{X}$ have been standardized. The i^{th} row of $\mathbf{X_k}$ is denoted by $\mathbf{x_{ik}}$. Assuming the logistic link function for F in (3.1), the goal of the model selection procedure proposed in

this chapter is to identify sparse regression models that have high predictive probability. I propose to do this by identifying the highest posterior probability model $\mathbf{k}$ for data $\mathbf{y}$, distributed according to

$$y_i \,|\, \boldsymbol{\beta}_{\mathbf{k}} \sim \text{Bernoulli}\left[\frac{\exp(\mathbf{x}'_{i\mathbf{k}}\boldsymbol{\beta}_{\mathbf{k}})}{1+\exp(\mathbf{x}'_{i\mathbf{k}}\boldsymbol{\beta}_{\mathbf{k}})}\right], \tag{3.2}$$

under prior constraints on the model space and the assumption of nonlocal prior density constraints on the regression parameter $\boldsymbol{\beta}_{\mathbf{k}}$. My primary focus is on the case $p \gg n$.

Recall from Chapter 2, Bayesian model selection is based on the calculation of posterior model probabilities. The posterior probability of model $\mathbf{j} \in \mathcal{J}$ and the marginal probability of the data under model $\mathbf{k}$ were calculated in (2.4) and (2.5), respectively.

The art in implementing a Bayesian model selection procedure thus focuses on specifying the prior densities $\pi_{\mathbf{k}}(\boldsymbol{\beta}_{\mathbf{k}})$ for $\boldsymbol{\beta}_{\mathbf{k}}$ under each model, as well as the prior model probabilities $p(\mathbf{k})$ for the models themselves. Except for the intercept, I assume nonlocal priors on the components of the regression vector in each model. For more information on this refer to Section 2.2 in Chapter 2.

3.2.1 Nonlocal Priors

The form of the nonlocal prior densities imposed on the (non-zero) regression coefficients $\boldsymbol{\beta}_{\mathbf{k}}$ in this chapter take the form of a product of independent iMOM priors, or piMOM densities, expressible as

$$\pi(\boldsymbol{\beta}_{\mathbf{k}}|\tau, r) = \frac{\tau^{rk/2}}{\Gamma(r/2)^k}\prod_{i=1}^{k}|\beta_i|^{-(r+1)}\exp\Big(-\frac{\tau}{\beta_i^2}\Big). \tag{3.3}$$

Here $\boldsymbol{\beta}_{\mathbf{k}}$ a vector of coefficients of length k, and $r, \tau > 0$. Following what was discussed in section (2.2) for iMOM priors, the hyperparameter τ represents a scale parameter that determines the dispersion of the prior around $\mathbf{0}$, while r is similar to the shape parameter

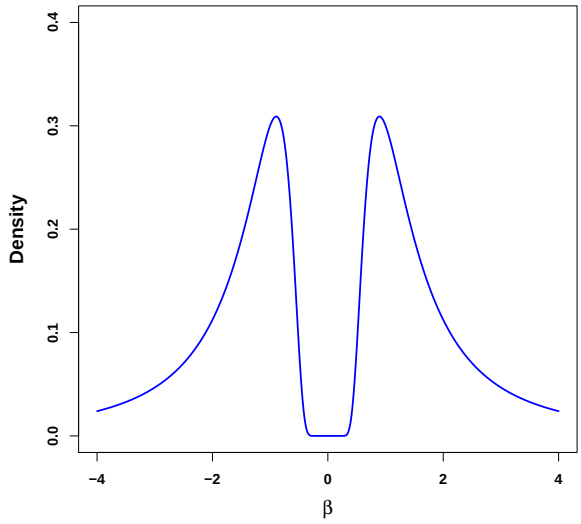


Figure 3.1: piMOM prior for $r = 1.5$ and $\tau = 1$

in Inverse Gamma distribution and determines the tail behavior of the density. An example of piMOM density is illustrated in Figure 3.1 for the particular case of $r = 1.5$ and $\tau = 1$.

3.2.2 Prior on Model Space

Following from my discussion in Section (2.3.2), I choose a beta-binomial prior as the prior for model size. The formulation specifies that the prior probability for model $\mathbf{k}$ is

$$p(\mathbf{k}) = \frac{B(a+k, b+p-k)}{B(a,b)}, \tag{3.4}$$

where $B(a,b)$ denotes the beta function and a and b are prior parameters that describe an underlying beta distribution on the marginal probability that a selected feature is associated with a non-zero regression coefficient in (3.2). This type of prior on the model size is also recommended in Castillo et al. (2015), where it is suggested that an exponential decrease in prior probabilities with model size provides optimal results when the prior density on

regression parameters has the form of a double exponential.

To incorporate my belief that the optimal predictive models are sparse, I arbitrarily set $a = \min(k^*, \lfloor \log(p) \rfloor)$, and $b = p - a$. For large n, this implies that I expect, on average, a feature vectors to be included in the model. Here, for the cases that $p/n > 4$, I pick $k^* = \underset{k}{\operatorname{argmax}} \binom{p}{k} < 2^n$, otherwise $k^* = 8$. This choice of k^* for the prior hyperparameter reflects the belief that the number of models that can be constructed from available covariates should be smaller than the number of possible binary responses. Similarly, by restricting a to be less than $\log(p)$, comparatively small prior probabilities are assigned to models that contain more than $\log(p)$ covariates. Finally, I impose a deterministic constraint on model size and define $P(\mathbf{k}) = 0$ if $k > n/2$.

A sensitivity analysis for a and b in (3.4) is provided in Section 3.4.1.1.

3.2.2.1 Choosing Hyperparameters

The algorithm for selecting hyperparamters is discussed in details in section 2.4 in chapter (2) and I employ that algorithm here. Notice that for a fixed p, the dispersion of the null distribution of the MLE around 0 decreases as the sample size n increases, although the rate of decrease is also affected by the structure of the design matrix $\mathbf{X}$. This makes the value of τ to decrease in order to maintain a fixed overlap threshold. This effect is illustrated in Table 3.1.

I note that a similar procedure for setting the scale parameter for local priors on the regression coefficients could potentially be implemented. Unfortunately, the application of this procedure to local priors can require extremely large values of the tuning parameters in order to "squash" the prior near 0 and achieve small overlap with the null distribution. As a consequence of this fact, the tuning parameters selected by this procedure will not reflect any reasonable prior belief on the values of the regression parameters in a logistic model with a standardized design matrix.

Ideally, I would adjust τ for each individual model, but as mentioned earlier it was not computationally feasible to do so for the applications and simulations reported in this chapter.

3.3 Numerical Aspects of Implementation

The model described in section 3.2 leads to a joint density for the data, model $\mathbf{k}$ and its parameters. As a result, the posterior distribution of model $\mathbf{k}$ and its coefficients can be expressed as

$$\begin{aligned}\pi(\boldsymbol{\beta}_{\mathbf{k}}, \mathbf{k}|\mathbf{y_n}) \propto \frac{\tau^{rk/2}}{\Gamma(r/2)^k} \prod_{i=1}^{k} |\beta_i|^{-(r+1)} \exp\left(-\frac{\tau}{\beta_i^2}\right) \times \\ \frac{B(a+k, b+p-k)}{B(a,b)} \prod_{j=1}^{n} \left\{\frac{e^{\mathbf{x}_{j\mathbf{k}}^T \boldsymbol{\beta}_{\mathbf{k}}}}{1+e^{\mathbf{x}_{j\mathbf{k}}^T \boldsymbol{\beta}_{\mathbf{k}}}}\right\}^{y_j} \left\{\frac{1}{1+e^{\mathbf{x}_{j\mathbf{k}}^T \boldsymbol{\beta}_{\mathbf{k}}}}\right\}^{1-y_j}.\end{aligned} \tag{3.5}$$

Because of the high dimension of the parameter space and the complexity of the posterior density function in (3.5), it is not feasible to maximize this function analytically to obtain the HPPM. To search for the HPPM, I therefore utilized a Markov chain Monte Carlo algorithm. To reduce the dimension of the parameter space, I used a Laplace approximation to marginalize over the regression coefficient $\boldsymbol{\beta}_{\mathbf{k}}$ associated with each model. The resulting approximation to the marginal posterior density of the data $\mathbf{y}$ under model $\mathbf{k}$ can be expressed as

$$\begin{aligned}m_{\mathbf{k}}(\mathbf{y_n}) = \int \pi(\mathbf{y_n}|\boldsymbol{\beta}_{\mathbf{k}}) \pi_{\mathbf{k}}(\boldsymbol{\beta}_{\mathbf{k}}) d\boldsymbol{\beta}_{\mathbf{k}} \approx \\ (2\pi)^{\frac{k}{2}} |\Sigma|^{-\frac{1}{2}} \pi(\mathbf{y_n}|\tilde{\boldsymbol{\beta}}_{\mathbf{k}}) \pi_{\mathbf{k}}(\tilde{\boldsymbol{\beta}}_{\mathbf{k}}).\end{aligned} \tag{3.6}$$

Here $\tilde{\boldsymbol{\beta}}_{\mathbf{k}}$ is the MAP estimate of $\boldsymbol{\beta}_{\mathbf{k}}$ and $|\Sigma|$ is the determinant of the Hessian of the function $f(\mathbf{y_n}, \boldsymbol{\beta}_{\mathbf{k}}) = -\log(\pi(\mathbf{y_n}|\boldsymbol{\beta}_{\mathbf{k}})) - \log(\pi_{\mathbf{k}}(\boldsymbol{\beta}_{\mathbf{k}}))$, computed at $\tilde{\boldsymbol{\beta}}_{\mathbf{k}}$. For more information

on Laplace approximation refer to section 2.3.1.

The elements of the Hessian matrix can be expressed as

$$H_{i,j}(\boldsymbol{\beta}_{\mathbf{k}}) = \begin{cases} i = j; & -\frac{r+1}{\beta_{ik}^2} + 6\tau\beta_{ik}^{-4} + \sum_s \frac{x_{si}^2 e^{\mathbf{x}'_{\mathbf{sk}}\boldsymbol{\beta}_{\mathbf{k}}}}{(1+e^{\mathbf{x}'_{\mathbf{sk}}\boldsymbol{\beta}_{\mathbf{k}}})^2} \\ i \neq j; & \sum_s \frac{x_{si}x_{sj}e^{\mathbf{x}'_{\mathbf{sk}}\boldsymbol{\beta}_{\mathbf{k}}}}{(1+e^{\mathbf{x}'_{\mathbf{sk}}\boldsymbol{\beta}_{\mathbf{k}}})^2} \end{cases} . \tag{3.7}$$

A simple birth-death scheme was used to sample from the posterior distribution. At each iteration of MCMC algorithm, each of the p covariates was visited in random order. The update at position i was performed by proposing a candidate model by flipping the inclusion state of that variable in the model. The candidate model was accepted using a Metropolis algorithm where the probability of accepting the candidate model, $\mathbf{k}^{\text{cand}}$, was

$$r = \frac{m_{\mathbf{k}^{\text{cand}}}(\mathbf{y_n})p(\mathbf{k}^{\text{cand}})}{m_{\mathbf{k}^{\text{curr}}}(\mathbf{y_n})p(\mathbf{k}^{\text{curr}})}. \tag{3.8}$$

The MAP estimate for $\boldsymbol{\beta}_{\boldsymbol{k}}$ was obtained using the `nlminb()` function in R. I assumed that an intercept was present in all models.

3.3.1 Convergence Diagnostics

Convergence diagnostics of MCMC can be used to assess whether an adequate number of iterations have been performed. Because of the high dimension of the parameter space for even moderately large p, I implemented a modified coupling diagnostic (Johnson, 1996, 1998) to assess the probability that my MCMC algorithm had identified the true model. In the standard implementation of this method, one randomly initializes two MCMC chains by independently including each variable in the model according to a fixed probability. The components of the model in each chain are then updated synchronously, using the same uniform random deviate to perform acceptance/rejection of the candidate models. The chains are said to couple when the models from each chain are identical. Note that

once the chains become coupled, they never uncouple. In theory, the distribution of the number of updates of the chains required to obtain coupling can be used to establish a bound on the Total Variation Distance (TVD) between iterates in the chain and the target distribution.

In my implementation of the coupling diagnostic, I started 100 pairs of model chains. Each pair was updated until either they had coupled or all p components in each of the chains had been updated N times, where $N = 250$. The (local) HPPM identified by each chain was recorded, and then the HPPM's for the 100 chains were compared. I then identified the global HPPM among the 100 models in the paired chains, and also examined the proportion of chains that had both coupled and identified the "global" HPPM. If the proportion was not high enough, it was possible that the paired chains required more updates to reach stationary distribution. This can be checked by increasing the number of updates, N.

This kind of implementation was proposed to overcome a potential convergence issue. Depending on the design matrix, there could be some pairs in which the final model in one chain is different from the final model in the other but the selected models span the same subspace. In this case, the paired chains are never coupled despite converging to same subspace.

3.4 Results

To investigate the performance of the proposed model selection procedure, I applied my procedure to both simulated data sets and real data. I compared the performance of my algorithm to ISIS-SCAD (Fan and Lv, 2008) in both real and simulated data because ISIS-SCAD has proven to be among the most successful model selection procedures used in practice. For the real data analyses, I also compared my method to another Bayesian procedure based on the product moment prior (Rossell et al., 2013).

3.4.1 Simulation Studies

In all simulation studies, I assumed that the response vector represents a sequence of Bernoulli samples whose component probabilities of success are given by

$$\pi_i = \frac{e^{\mathbf{x_{ik}}^T \boldsymbol{\beta_k}}}{1 + e^{\mathbf{x_{ik}}^T \boldsymbol{\beta_k}}} \tag{3.9}$$

for a true model **k**.

Elements of the design matrix $\mathbf{X}$ were sampled from a multivariate normal distribution with mean 0 and covariance matrix Σ, where the diagonal elements of Σ were 1 and off diagonal elements were 0.5. That is,

$$\Sigma = \begin{pmatrix} 1 & 0.5 & \cdots & 0.5 \\ 0.5 & 1 & \cdots & 0.5 \\ \vdots & \vdots & \ddots & \vdots \\ 0.5 & 0.5 & \cdots & 1 \end{pmatrix}_{p \times p} \tag{3.10}$$

$\mathbf{x_j} \sim N_p(0, \Sigma)$; $\mathbf{x_j}$: j^{th} row of design matrix $\mathbf{X}$

$y_i \sim$ Bernoulli(π_i).

Different combinations of n and p were investigated. Moreover, different ranges of regression coefficients were tested. In my simulations, the true model contained 3 variables. The goal was to find the true model, as well as estimating corresponding coefficients.

The following combinations of n, p and $\boldsymbol{\beta}$ were used to perform the simulation studies which covers variety of combinations between sample size and number of covariates.

- $n \in \{50, 100, 200, 400, 600\}$
- $p \in \{1000, 10000\}$

Table 3.1: Selected τ parameter of piMOM prior for different simulation settings

	$n = 50$	$n = 100$	$n = 200$	$n = 400$	$n = 600$
$p = 1000$	5.50	1.66	0.68	0.30	0.20
$p = 10,000$	4.28	1.85	0.76	0.34	0.21

Table 3.2: Selected r parameter of piMOM prior for different simulation settings

	$n = 50$	$n = 100$	$n = 200$	$n = 400$	$n = 600$
$p = 1000$	2.04	1.50	1.24	1.07	1.00
$p = 10,000$	1.90	1.54	1.27	1.09	1.01

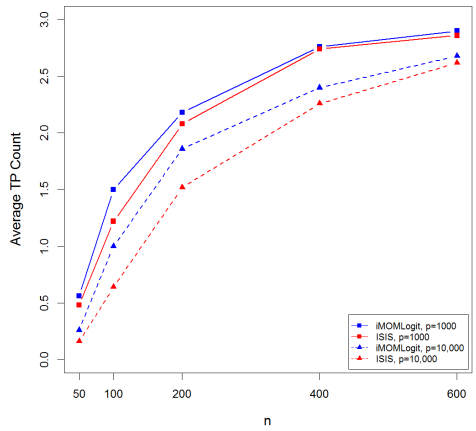
- $\boldsymbol{\beta} \in \{\boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \boldsymbol{\beta_3}\}$, where the non-zero coefficients of the $\boldsymbol{\beta_i}$ vector were the i^{th} row of the matrix $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 4 & 5 & 6 \end{bmatrix}$.

The hyperparameters τ and r for the piMOM prior were selected by the procedure explained in Section 3.2.2.1 for each of the 10 combinations of n and p. Values of τ and r selected by this procedure are summarized in Table 3.1 and 3.2 respectively.

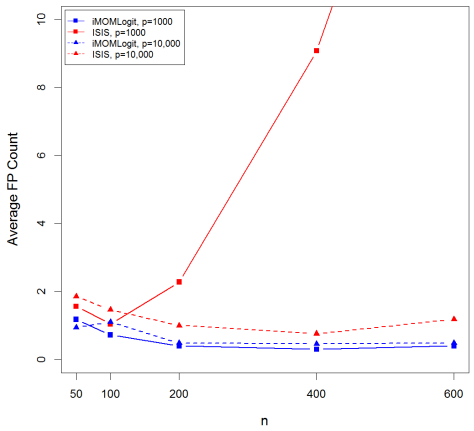
To run ISIS-SCAD, I used the R package *SIS* (Fan et al., 2015) available from CRAN.

The variable selection procedure in both algorithms was run 50 times for each of the 30 combinations of n, p and $\boldsymbol{\beta}$. In each trial, true and false positive values for iMOMLogit and ISIS-SCAD were counted by comparing the selected model with the true one. TP and FP rates were defined as the average true and false positive values over 50 trials. A true positive, TP, was defined to be the number of variables that were correctly selected, while false positives, FP, were the number of variables that were mistakenly selected.

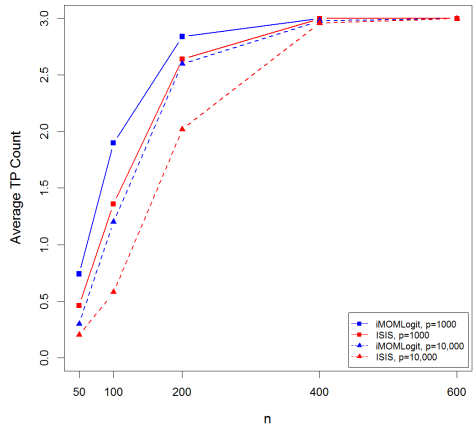
Figure 3.2 shows average TP and FP counts of both methods for all combinations of n and p and $\boldsymbol{\beta}$. I see that all follow the same trend. In all cases, the average FP count for iMOMLogit was less than ISIS-SCAD, while its average TP count was higher. The only



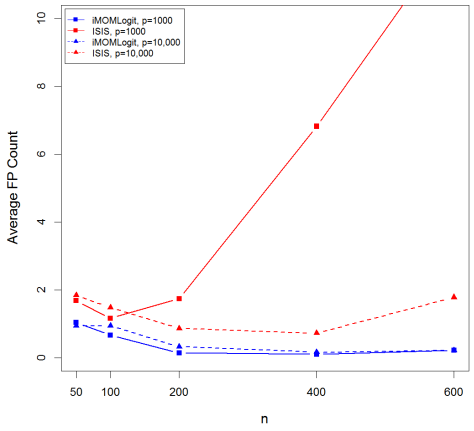
Average true positive count for $\boldsymbol{\beta_1}$



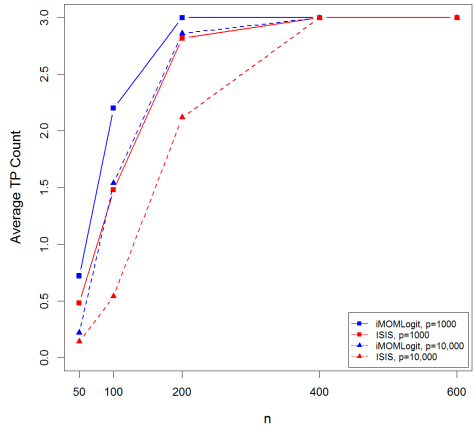
Average false positive count for $\boldsymbol{\beta_1}$



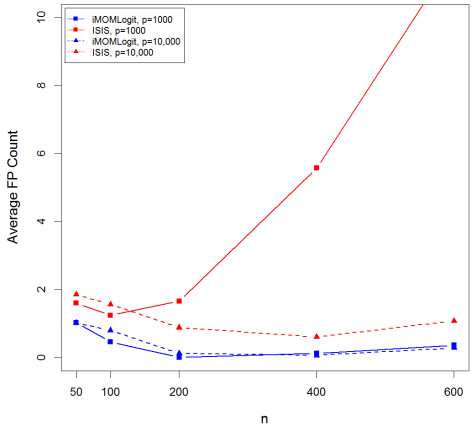
Average true positive count for $\boldsymbol{\beta_2}$



Average false positive count for $\boldsymbol{\beta_2}$



Average true positive count for $\boldsymbol{\beta_3}$



Average false positive count for $\boldsymbol{\beta_3}$

Figure 3.2: Average true and false positive counts for all 30 different simulation settings.

case where both iMOMLogit and ISIS-SCAD had the same average TP count was when they both found the true model in all 50 simulation trials.

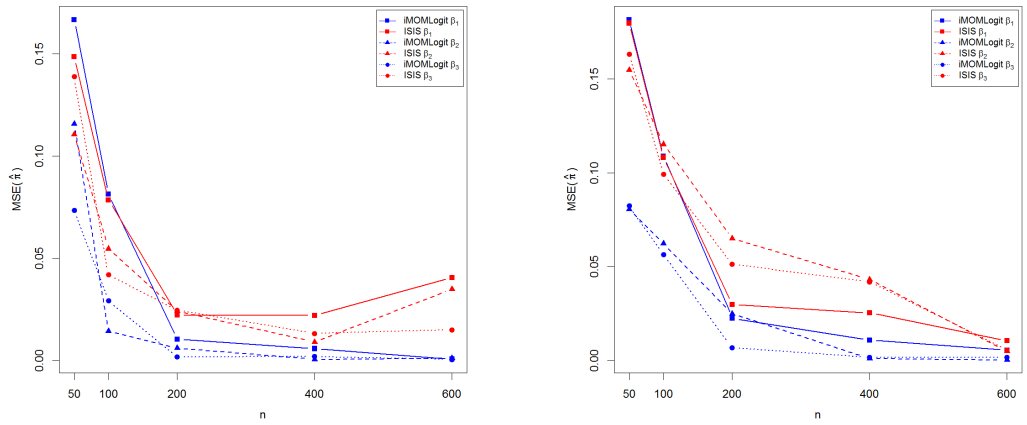
I next compared the performance of both methods in estimating the regression coefficients. For each simulation setting, I compared the mean squared error in estimating the probability of success for each binary observation by performing 10-fold cross validation. The point estimate $\hat{\boldsymbol{\beta}}$ was estimated as the posterior mode under the HPPM. The predicted value of $\hat{\pi}$ was then computed according to (3.1). Note that the prediction of the response vector involves both coefficient estimation and variable selection. The mean squared error of prediction (MSE) was defined as follows:

$$\text{MSE}(\hat{\boldsymbol{\pi}}) = \frac{1}{n}\|\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}\|^2 = \frac{1}{n}\sum_{i=1}^{n}(\hat{\pi}_i - \pi_i)^2. \tag{3.11}$$

The comparison between cross validated MSEs of both methods is shown in Figure 3.3. As in the comparisons of TP and FP rates, these figures suggest that iMOMLogit is preferred to ISIS-SCAD in estimating the success probabilities of binary observations.

3.4.1.1 Sensitivity Analysis for Prior Parameters on Model Space

To assess the sensitivity of my results to the prior hyperparameters on the model space (3.4), I conducted a brief sensitivity analysis under the simulation settings for which $n = 200$, $p = 1000$ and $\boldsymbol{\beta} = [4, 5, 6]^T$. I also fixed $b = p - a$ as in my entire analysis. This insured that the prior mean of the number of variables selected would be a. Based on the default procedure for defining a described in Section 3.2.2, the default value for a in this setting was 6. I examined sensitivity to this choice of a by varying a around this default value within the interval $(3, 9)$. To quantitatively assess the sensitivity of the selection procedure to values of a in this range, I examined the consequent changes to $\text{MSE}(\hat{\boldsymbol{\pi}})$ described in (3.11). This measure incorporates errors in both variable selection and coefficient estimation.



$p = 1000$ $p = 10,000$

Figure 3.3: 10-fold cross validation MSE($\hat{\boldsymbol{\pi}}$) of iMOMLogit vs. ISIS-SCAD, for $p = 1000$ and $p = 10,000$.

The Figure 3.4 depicts MSE($\hat{\boldsymbol{\pi}}$) for different values of a in the described simulation setting. The output at nominal value of a is indicated by a red square. As can be seen, the value of output does not change dramatically with changes in a, varying by at most 4.8×10^{-5} from the default choice of a.

3.4.2 Real Data Analysis

I applied iMOMLogit to two data sets, one with a small sample size and one with a large sample size. These two data sets are publicly available and have good clinical annotations. The first data set is the Golub leukemia data (Golub et al., 1999). The goal of my analysis for these data was is to discriminate between two types of acute leukemia, myeloid (AML) and lymphoblastic (ALL). The design matrix consists of gene expression levels produced by cDNA microarrays from bone marrow samples, pre-processed by RMA (Irizarry et al., 2003). There are 72 samples and 7,129 genes in the data set. The second data set is the clear cell Renal Cell Carcinoma (ccRCC) RNAseq data available from the

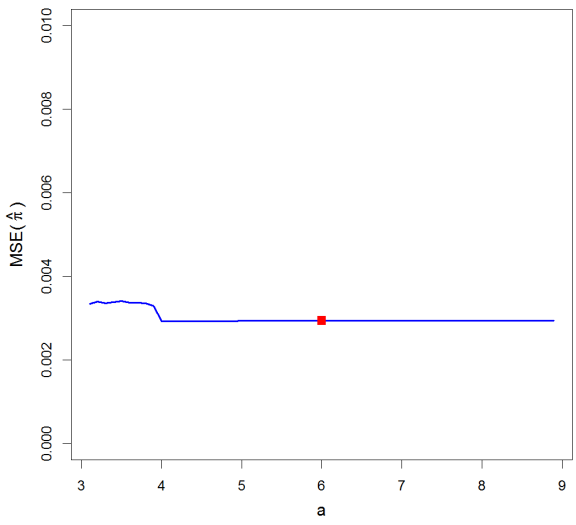


Figure 3.4: Sensitivity analysis for parameters of prior on model space.

Cancer Genome Atlas projects (Cancer Genome Atlas Research Network, 2013) (TCGA). There are 467 tumor samples and more than 20,000 genes as features in this data set.

As mentioned earlier, I also compared my selection procedure results to a related Bayesian method proposed in Rossell et al. (2013), called pmomPM. This method uses a probit link function with a moment prior, (pMOM), another type of nonlocal prior. The pMOM prior has Gaussian tails and decreases quadratically near the origin. I implemented this method with the default hyperparameter suggested in Rossell et al. (2013) for sparse models. To run pmomPM method, I used the R package 'mombf' (Rossell et al., 2015) available from CRAN.

In contrast to iMOMLogit and ISIS-SCAD, the mombf package focuses on prediction using Bayesian model averaging, rather than on the identification of biologicially important genes using the HPPM. Because of the behavior of the pMOM prior near the origin, the pMOM model selects many more genes in the models over which it averages. Though

model averaging can improve predction accuracy (Raftery et al., 1997), the current version of mombf package does not provide estimates of the HPPM, which complicates comparisons with the other methods considered here. These attributes of the pmomPM method are illustrated in the examples that follow.

3.4.2.1 Leukemia Data

Following Golub et al. (1999), I split the data into training and test sets. The training set contains 38 samples, with 27 ALL and 11 AML. The testing set contains 34 samples, with 20 ALL and 14 AML.

Table 3.3 summarizes the results of applying iMOMLogit, ISIS-SCAD and pmomPM to these data. The error rate for predicting the test data observations was 5.88% for iMOMLogit, which misclassified 2 out of 34 observations, samples 17 and 31. Both ISIS-SCAD and the method described in Golub et al. (1999) resulted in an error rate of 14.7%. ISIS-SCAD achieved this error rate by finding two significant genes, 'Zyxin' and 'FAH', whereas Golub et al. (1999) selected 50 genes. The pmomPM method achieved an error rate of 23.53% with an average model size of 11.08. None of the genes were assigned marginal posterior probability of 0.5 by the pmomPM method; the highest marginal posterior probability of any gene was 0.052, acheived by CD33.

iMOMLogit selected a model containing only one gene named 'Zyxin' , which perfectly predicted the classifications in the training data. This gene was also listed in the top 50 genes reported by Golub et al. (1999), and was found to be advantageous for classifying the two types of leukemia in four published data sets (Baker and Kramer, 2006). The gene 'FAH' found only by ISIS-SCAD is involved in certain metabolic pathways that are not known to be associated with leukemia (Kegg.org).

Following the methodology discussed in section 3.3.1, 74% of pairs of chains that were updated using the coupling algorithm found the same highest posterior probability model

Table 3.3: Comparison between iMOMLogit and other methods for leukemia data set

Method	Error Rate	Reported Genes
iMOMLogit	5.88%	Zyxin
ISIS-SCAD	14.70%	Zyxin - FAH
pmomPM	23.53%	No genes had marginal posterior probability greater than 0.5

(HPPM). Among all pairs, 95% coupled.

3.4.2.2 Renal Cell Carcinoma Data

This data set was generated by the Cancer Genome Atlas Research Network (2013) and contains Illumina HiSeq data on mRNA expression for 467 patient samples. The survival outcomes of these patients were available. A hierarchical clustering of the gene expression data (preprocessed using DeMix (Ahn et al., 2013) to remove stromal contamination) were performed on the data. That led to the identification of four clusters of patients based on survival times. To apply iMOMLogit, I considered two of those clusters, presenting the best and worst survival outcomes and labeled them as 0 (worst) and 1 (best). The resulting number of samples included in my analysis was 193, with 14,150 features in the design matrix.

The results using iMOMLogit, ISIS-SCAD and pmomPM are summarized in Table 3.4. To compare methods, I performed a 10-fold cross-validation. The error rate of iMOMLogit was 9.79%, ISIS-SCAD's error rate was 12.97%, and pmomPM was 9.84%. In the model selected by iMOMLogit, there were 3 significant genes named 'C7orf43', 'NUMBL' and 'SAV1', with the latter two being uniquely identified by my model. 'NUMBL' participates in the Notch signaling pathway and is believed to contribute to nervous system tumors (glioma) (Tao et al., 2012) as well as lung cancer (Yingjie et al., 2013). Notch signaling pathway is highly conserved, manages communication between adjacent cells

and maintenance of adult stem cells, and is linked to the development of various cancers (Alketbi and Attoub, 2015). Not surprisingly, I identified NUMBL as differentiating two groups of kidney patients. 'SAV1' has been reported to play a role in kidney cancer (Matsuura et al., 2011), and is located in a Hippo signaling pathway (Kegg.org). The Hippo signaling pathway is highly conserved and controls epithelial tissue growth. Recently, its relation to other signaling pathways has been studied to identify new therapeutic interventions for cancer (Yimlamai et al., 2015).

Among all pairs of chains with different random starts, 32% of them reported the same global HPPM and 6% of paired chains were coupled. This suggests that convergence in this data set was more problematic, and that my multiple coupled chain approach, or other modifications of the standard, single chain MCMC algorithm, is required to identify the HPPM model.

The genes uniquely selected by ISIS-SCAD were 'C19orf66', 'ATXN7L2' and 'MICAL1'. 'ATXN7L2' was previously reported to be associated with non-small cell lung cancer (Wu et al., 2013), whereas 'MICAL1' was previously reported to control survival in melanoma cell lines.

As for the leukemia data, the pmomPM selected substantially more genes in each of its sampled models, and the genes selected in each model were highly variable. The average model size of the pmomPM method for this data set was 13.84. As before, none of the genes were assigned marginal probability of 0.5; the highest marginal posterior probability assigned to any gene was 0.33, for API5.

The genes identified by iMOMLogit seem to be more biologically meaningful and better annotated in the literature for ccRCC than those selected by ISIS-SCAD.

Table 3.4: Comparison between iMOMLogit and other methods for renal cell carcinoma data set

Method	Error Rate	Reported Genes
iMOMLogit	9.79%	C7orf43 - NUMBL - SAV1
ISIS-SCAD	12.97%	C7orf43 - C19orf66 - ATXN7L2 - MICAL1
pmomPM	9.84%	No genes had marginal posterior probability greater than 0.5

3.5 Discussion

In this chapter I introduced a Bayesian method, iMOMLogit, for variable selection in binary response regression problems in high and ultra-high dimensional settings. There are many applications associated with these type of data. Such data are of great interest to bioinformaticians and biologists, who routinely collect gene expression data to find prognostic features to classify cancer types.

In such classification problems where the goal is to find the true significant features while keeping the false positives as low as possible, having higher precision that is defined by $\frac{TP}{TP+FP}$ will be an advantage for a selection method. As shown in Figure 3.2, iMOMLogit shows a high precision in finding true variables compared to ISIS-SCAD with a higher FP counts which make it less precise in the simulation analysis. In predicting the success probabilities of the binary response vector, iMOMLogit's demonstrates a very good performance compared to ISIS-SCAD as illustrated in Figure 3.3.

For two real datasets, iMOMLogit identified sparse models with low prediction error rates. In both cases, biological considerations suggest that the genes reported by iMOMLogit appear to be valid predictors of biological outcomes.

The primary disadvantage of the iMOMLogit procedure is that it is computationally much more intensive than ISIS-SCAD and related penalized likelihood methods. However, by implementing the whole algorithm in C++ language the procedure is fairly fast.

This is discussed in details in chapter 6 where I introduce an R package containing the algorithms proposed in this dissertation.

4. HIGH DIMENSIONAL BAYESIAN VARIABLE SELECTION FOR SURVIVAL DATA

4.1 Introduction

Recent developments in gene sequencing technology have made it easier to generate massive genomic data that can be used to make new discoveries in genomics. The outcomes for most cancer studies are survival times for subjects, and the goal is to investigate the relation or any potential association between survival times and the covariates in the model; namely, genes in this context.

Survival times for each subject represent either the time to death or disease progressoin, or the time to study termination or the time until the subject is lost to follow up. In the latter cases, the subject's survival time is *censored*. The relation between survival times and covariates is modeled through the conditional hazard function, which is the probability of death in the interval $(t, t + \Delta t)$ when Δt becomes really small, given the covariates in the study. A more preciese definition is

$$h(t|\mathbf{X}) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} P(t \leq T \leq t + \Delta t \,|\, T \geq t, \mathbf{X}). \tag{4.1}$$

Here, $\mathbf{X} = (X_1, X_2, \ldots, X_p)$ is the $n \times p$ design matrix with n observations and p covariates. Proportional hazard models are of the form

$$h(t \,|\, \mathbf{X}) = h_0(t)\Phi(\mathbf{X}). \tag{4.2}$$

with identifiability constraint of $\Phi(\mathbf{0}) = 1$. In this formula, $h_0(t)$ denotes the baseline hazard function. One special case of such model is the one proposed by Cox (1972) where $\Phi(\mathbf{X}) = \exp\{\mathbf{X}^T\boldsymbol{\beta}\}$. Consequently the hazard function in Cox proportional hazard model

is obtained by

$$h(t \mid \mathbf{X}) = h_0(t)e^{\mathbf{X}^T\boldsymbol{\beta}}. \tag{4.3}$$

Here, $\boldsymbol{\beta}$ is $p \times 1$ vector of coefficients. In this model, estimating $\boldsymbol{\beta}$ does not depend on $h_0(t)$. Moreover, this term is canceled out in comparing marginal probabilities of data under different models in the Bayesian variable selection procedure. For general survival analysis, however, the baseline hazard function is necessary for predicting survival times and can be estimated non-parametrically. For more information, interested readers can consult Cox and Oakes (1984); Kalbfleisch and Prentice (2002).

Gene expression datasets are usually in ultrahigh dimensions where thousands of genes are examined for only hundreds of subjects. However, only a limited number of genes contribute significantly to the outcome. In other words, most of the elements in the vector $\boldsymbol{\beta}$ are zero. This is the sparsity assumption imposed in variable selection problems. The primary target is then to find covariates with non-zero coefficients or, equivalently, those genes that contribute the most in determining the survival outcome.

As discussed in Chapter 1, most common classical penalized likelihood approaches have been extended for survival data. This includes LASSO (Tibshirani et al., 1997), adaptive LASSO (Zhang and Lu, 2007), Dantzig selector (Antoniadis et al., 2010) and ISIS-SCAD (Fan and Li, 2002; Fan et al., 2010). Some Bayesian methods have also been introduced for this problem. Faraggi and Simon (1998) define a Gaussian prior on vector of coefficients and a loss function in order to select a parsimonious model. Sha et al. (2006) exploit spike-and-slab prior similar to what was proposed by George and McCulloch (1997). Ibrahim et al. (1999) use a multivariate Gaussian distribution for the coefficient vector in the selection process.

All of the aforementioned Bayesian methods use local priors for model coefficients.

In this chapter, as an extension to the previous work for logistic regression (Nikooienejad et al., 2016) discussed in Chapter 3, I propose a Bayesian method based on a mixture prior of a point mass at zero and a nonlocal prior. In particular, inverse moment prior for elements of $\boldsymbol{\beta}$ in (4.3) and find the model with highest posterior probability. The computationally burdensome MCMC process is avoided by adapting a stochastic search based method, S5 (Shin et al., 2015). A general algorithm is also provided to set the tuning parameter of the nonlocal prior.

This chapter is structured as follows. In Section 4.2 I introduce notation, review preliminary points, and discuss the methodology behind my proposed method. Section 4.3 provides simulation and real data analyses to illustrate the performance of the proposed method and compares it to several competing methods. Finally Section 4.4 concludes this chapter.

4.2 Methods

4.2.1 Preliminaries

Let T_i denote the survival and C_i denote the censoring times for individual i. Each element in the observed vector of survival times, $\mathbf{y}$, is defined as $y_i = \min\{T_i, C_i\}$. The status for each individual is defined as $\delta_i = I(T_i \leq C_i)$. The status vector is represented by $\boldsymbol{\delta} = (\delta_1, \delta_2, \ldots, \delta_n)^T$. I assume the censoring mechanism is at random, meaning that C_i and T_i are conditionally independent given $\mathbf{X}_i$, where $\mathbf{X}_i \in \mathbb{R}^p$ are the covariates for individual i, and comprise the i^{th} row of $\mathbf{X}$. The observed data is of the form $\big\{(y_i, \delta_i, \mathbf{X}_i);\ i = 1, 2, \ldots, n\big\}$.

Model $\mathbf{k}$ is defined as $\mathbf{k} = \{k_1, \ldots, k_j\}$ where $(1 \leq k_1 < \cdots < k_j \leq p)$ and it is assumed that $\beta_{k_1} \neq 0, \ldots, \beta_{k_j} \neq 0$ and all other elements of $\boldsymbol{\beta}$ are 0. The design matrix corresponding to model $\mathbf{k}$ is denoted by $\mathbf{X_k}$, and the regression vector by$\boldsymbol{\beta}_\mathbf{k} = (\beta_{k_1}, \beta_{k_2}, \ldots, \beta_{k_j})^T$.

Let $\mathcal{R}(t) = \{i : y_i \leq t\}$ represent the *risk set* at time t, the set of all individuals who are still present in the study at time t and are neither dead nor censored. I also assume throughout the chapter that the failure times are distinct. In other words, only one individual fails at a specific failure time. With this assumption and letting $\xi_{\mathbf{k}i} = \exp\{\mathbf{X}_{\mathbf{k}i}^T \boldsymbol{\beta}_{\mathbf{k}}\}$, the partial likelihood (Cox, 1972) for $\boldsymbol{\beta}_{\mathbf{k}}$ in model $\mathbf{k}$ can be written as

$$L_p(\boldsymbol{\beta}_{\mathbf{k}}) = \prod_{i=1}^{n} \left[\frac{\xi_{\mathbf{k}i}}{\sum_{j \in R(y_i)} \xi_{\mathbf{k}j}} \right]^{\delta_i}. \tag{4.4}$$

My method uses this partial likelihood as the sampling distribution in my Bayesian model selection scheme. This is a point of discussion as there might be some information loss in (4.4) with respect to $\boldsymbol{\beta}_{\mathbf{k}}$. For instance, Basu (Basu, 2012) argues that partial likelihoods can not usually be interpreted as sampling distributions. On the other hand, Berger et al. (1999) encourage the use of partial likelihoods when the nuisance parameters are marginalized out. I chose to test this idea and use the partial likelihood in (4.4) as if it was the sampling distribution for observed survival times.

Sorting the observed unique survival times in ascending order and consequently re-ordering the status vector $\boldsymbol{\delta}$ as well as the design matrix $\mathbf{X}$ with respect to the ordered $\mathbf{y}$, the sampling distribution of $\mathbf{y}$ for model $\mathbf{k}$ can be written as

$$\pi(\mathbf{y} \mid \boldsymbol{\beta}_{\mathbf{k}}) = \prod_{i=1}^{n} \left[\frac{e^{\mathbf{X}_{\mathbf{k}i}\boldsymbol{\beta}_{\mathbf{k}}}}{\sum_{j=i}^{n} e^{\mathbf{X}_{\mathbf{k}j}\boldsymbol{\beta}_{\mathbf{k}}}} \right]^{\delta_i}. \tag{4.5}$$

This is the sampling distribution that is used in the Bayesian hierarchical modeling discussed in section 2.3. Recall that computing the posterior probability for model $\mathbf{j}$ in that section was written as

$$p(\mathbf{j} \mid \mathbf{y}) = \frac{p(\mathbf{j}) m_{\mathbf{j}}(\mathbf{y})}{\sum_{k \in \mathcal{J}} p(\mathbf{k}) m_{\mathbf{k}}(\mathbf{y})}, \tag{4.6}$$

where $\mathcal{J}$ is the set of all possible models, $p(\mathbf{k})$ is the prior for model $\mathbf{k}$ and the marginal probability of the data under model $\mathbf{k}$ is computed by

$$m_{\mathbf{k}}(\mathbf{y}) = \int \pi(\mathbf{y} \mid \boldsymbol{\beta}_{\mathbf{k}}) \pi_{\mathbf{k}}(\boldsymbol{\beta}_{\mathbf{k}}) d\boldsymbol{\beta}_{\mathbf{k}}. \tag{4.7}$$

The prior density for $\boldsymbol{\beta}_{\mathbf{k}}$ and the prior on model space impact the overall performance of the selection procedure and the amount of sparsity imposed on candidate models. Note that the sampling distribution in (4.5) is continuous in $\boldsymbol{\beta}_{\mathbf{k}}$, and I define an inverse moment prior (Johnson and Rossell, 2010) on each of the coefficients in model $\mathbf{k}$.

For the prior on model space I use the beta-binomial prior, the same prior I used for logistic regression variable selection and investigated in detail in Chapter 2. That is,

$$p(\mathbf{k}) = \frac{B(a+k, b+p-k)}{B(a,b)}, \tag{4.8}$$

where $B(a,b)$ denotes the beta function and a and b are prior parameters that describe an underlying beta distribution on the marginal probability of model $\mathbf{k}$. I also showed in section 2.3.2 that by setting the hyperparameter $b = p - a$, the average prior size of the selected model is a with the variance $a - \frac{a^2}{p} \approx a$ for large values of p. Another setting for a and b could be $a = b = 1$, which results in a uniform-binomial prior.

To incorporate my belief that the optimal predictive models are sparse, I set $a = 1$ and $b = p - a$. By this structure, comparatively small prior probabilities are assigned to models that contain many covariates.

4.2.2 Product Inverse MOMent (piMOM) Prior

In my method, the form of the nonlocal prior densities imposed on the non-zero coefficients, $\boldsymbol{\beta}_{\mathbf{k}}$, take the form of a product of independent piMOM priors, or piMOM densities

(Johnson and Rossell, 2010), expressible as

$$\pi(\boldsymbol{\beta}_{\mathbf{k}}|\tau, r) = \frac{\tau^{rk/2}}{\Gamma(r/2)^k} \prod_{i=1}^{k} |\beta_i|^{-(r+1)} \exp\Big(-\frac{\tau}{\beta_i^2}\Big). \tag{4.9}$$

Here $\boldsymbol{\beta}_{\mathbf{k}}$ is a vector of coefficients of length k, and $r, \tau > 0$. The hyperparameter τ represents a scale parameter that determines the dispersion of the prior around $\mathbf{0}$, while r is similar to the shape parameter in Inverse Gamma distribution and determines the tail behavior of the density. For more details on this prior refer to section 2.2.

For selecting hyperparameters of the piMOM prior, I adopt the algorithm in section 2.4 where the null distribution of the maximum likelihood estimator for $\boldsymbol{\beta}_{\mathbf{k}}$ (when all components of $\boldsymbol{\beta}_{\mathbf{k}}$ are 0), obtained from a randomly selected design matrix $\mathbf{X}_{\mathbf{k}}$, is compared to the prior density on $\boldsymbol{\beta}_{\mathbf{k}}$ under the alternative assumption that the components are non-zero. I then choose a τ that makes the overlap between two densities less than a specified threshold, namely $1/\sqrt{p}$.

To generate the response vector under the null model the following procedure is performed. Under the null model, the survival times are sampled using the methodology proposed in Bender et al. (2005) under Cox-exponential distribution models, when all components of $\boldsymbol{\beta}$ are zero. As a result, for each individual, the sampled survival time under the null is computed as

$$t_i^s = -\frac{\log u_i}{\lambda_1 \exp\{\mathbf{X}\boldsymbol{\beta}\}} = -\frac{\log u_i}{\lambda_1}; \quad \text{where } u_i \sim U(0,1). \tag{4.10}$$

In this formulation, λ_1 is the baseline hazard function, $h_0(t)$, which I assume to be 1 in my analysis, and $U(0,1)$ is the uniform distribution between 0 and 1. I define the event rate to be the proportion of subjects that have $\delta_i = 1$. This can be estimated from observed data by taking the average of the event status vector, $\boldsymbol{\delta}$. Defining censoring rate as one minus

the even rate, The estimated censoring rate can be obtained by

$$\hat{c} = 1 - \hat{e}, \tag{4.11}$$

where $\hat{e}$ and $\hat{c}$ are the estimated event and censoring rate, respectively.

Let $\boldsymbol{t}^s$ and $\boldsymbol{c}^s$ be the vector of sampled survival times, and censoring times, respectively. The censoring times, are obtained independently by sampling from an exponential distribution with rate λ_2. The rate λ_2 is computed from the assumed rate for exponential distribution in sampling survival times in (4.10), λ_1, and the estimated censoring rate in observations, $\hat{c}$. More precisely, λ_2 is set so the censoring rate is equal to the observed censoring rate, $\hat{c}$. The details of this calculation are described by the following equtaion:

$$\hat{c} = \mathbb{E}\big[I(t_i^s > c_i^s)\big] = p(t_i^s > c_i^s) = \int_0^\infty \int_c^\infty \lambda_1 \lambda_2 e^{-\lambda_1 t} e^{-\lambda_2 c} dt dc = \frac{\lambda_2}{\lambda_1 + \lambda_2}. \tag{4.12}$$

Thus, letting $\lambda_1 = 1$, the rate λ_2 is computed as

$$\lambda_2 = \frac{\hat{c}}{1 - \hat{c}}, \tag{4.13}$$

which leads us to obtaining censoring times vector, $\boldsymbol{c}^s$. The sampled survivial time and status for each observation is then computed as

$$y_i^s = \min\{t_i^s, c_i^s\} \qquad \text{and} \qquad \delta_i^s = I(t_i^s \leq c_i^s), \tag{4.14}$$

which comprise $\mathbf{y}^s$ and $\boldsymbol{\delta}^s$ under the null model.

Using the pair $(\mathbf{y}^s, \boldsymbol{\delta}^s)$, the algorithm is exactly similar to the algorithm described in Section 2.4. Instead of using Maximum Likelihood Estimate (MLE) in the logistic

model, the MLE from Cox model is used. It should be noted that the distribution of the MLE for the Cox model under the null hypothesis is $\hat{\boldsymbol{\beta}} \sim \mathcal{N}\big(\mathbf{0}, I(\hat{\boldsymbol{\beta}})\big)$, where $I(\boldsymbol{\beta})$ is the information matrix of the partial likelihood function. Thus, it is appropriate to approximate the pooled estimated coefficients in that algorithm with a normal density function. When the sample size gets really large, the variance of the MLE decreases and causes the overlap to become really small and consequently small values of τ are selected.

In general, I found that $r = 1$ and $\tau = 0.25$ are good default values if one chooses not to run the hyperparameter selection algorithm. Those values are based on various simulation results that show reasonable behavior of the prior to cover small and fairly large coefficients. When $r = 1$, the peaks of the iMOM prior occurs at $-\sqrt{\tau}$ and $\sqrt{\tau}$. By equating $\sqrt{\tau}$ to the absolute value of the most common effect size for that application, it provides insight on what default value of τ would be for differnet applications. Nikooienejad et al. (2016) discuss the benefit of exploiting this algorithm for nonlocal priors compared to local ones.

4.2.3 Highest Posterior Probability Model

Computing the posterior probability for each model requires the marginal probability of observed survival times under each model as shown in (4.6), (4.7). The marginal probability is approximated by using the Laplace approximation where the regression coefficients in $\boldsymbol{\beta}_{\mathbf{k}}$ are integrated out. Details are discussed in Section 2.3.1 using the marginal probability defined in (2.13). This leads to

$$m_{\mathbf{k}}(\mathbf{y}_n) = \pi(\mathbf{y}_n \,|\, \hat{\boldsymbol{\beta}}_{\mathbf{k}})\pi(\hat{\boldsymbol{\beta}}_{\mathbf{k}})(2\pi)^{d/2}|G_{\hat{\boldsymbol{\beta}}_{\mathbf{k}}}|^{-1/2}.$$

Here, $\hat{\boldsymbol{\beta}}_{\mathbf{k}}$ is the maximum a posteriori (MAP) estimate of $\boldsymbol{\beta}_{\mathbf{k}}$ and $G_{\hat{\boldsymbol{\beta}}_{\mathbf{k}}}$ is the Hessian of the negative of the log posterior function, $g(\boldsymbol{\beta}_{\mathbf{k}}) = -\log(\pi(\mathbf{y} \,|\, \boldsymbol{\beta}_{\mathbf{k}})) - \log(\pi(\boldsymbol{\beta}_{\mathbf{k}}))$, computed at $\hat{\boldsymbol{\beta}}_{\mathbf{k}}$. Finding the MAP of $\boldsymbol{\beta}_{\mathbf{k}}$ is equivalent to finding the minimum of $g(\boldsymbol{\beta}_{\mathbf{k}})$

function.

4.2.3.1 Calculating the Gradient and Hessian of $g(\boldsymbol{\beta}_{\mathbf{k}})$

Let $l(\mathbf{y};\boldsymbol{\beta}_{\mathbf{k}}) = \log(\pi(\mathbf{y}\,|\,\boldsymbol{\beta}_{\mathbf{k}}))$ and $l_{\pi}(\boldsymbol{\beta}_{\mathbf{k}}) = \log(\pi(\boldsymbol{\beta}_{\mathbf{k}}))$. For an $n \times p$ matrix $\mathbf{A}$, let $\mathbf{A}_{(i)}$ denote the $n \times 1$ vector corresponding to the i^{th} column of $\mathbf{A}$ and $\mathbf{A}_j$ denote the $1 \times p$ vector corresponding to the j^{th} row of $\mathbf{A}$. Also let $\overline{\mathbf{A}}_i = (\mathbf{A}_{i:n,\cdot})^T$, where $\mathbf{A}_{i:n,\cdot}$ is the sub-matrix of $\mathbf{A}$ from row i to the last row where all columns are included. This makes the dimension of $\overline{\mathbf{A}}_i$ to be $p \times (n-i+1)$. Similarly, for a vector $\boldsymbol{\alpha}$ of size n, let $\overline{\boldsymbol{\alpha}}_i$ denote the sub-vector of $\boldsymbol{\alpha}$ from i^{th} element to the last one, a vector of size $(n-i+1)$.

Let $\psi_{\mathbf{k}_i} = \sum_{j=i}^{n} e^{\mathbf{X}_{\mathbf{k}j}\boldsymbol{\beta}_{\mathbf{k}}}$ and $\boldsymbol{\psi}_{\mathbf{k}} = (\psi_{\mathbf{k}_1}, \ldots, \psi_{\mathbf{k}_n})^T$. Also let $\boldsymbol{\eta}$ denote the $n \times 1$ column vector $\exp\{\mathbf{X}_{\mathbf{k}}\boldsymbol{\beta}_{\mathbf{k}}\}$. The logarithm of $\pi(\mathbf{y}\,|\,\boldsymbol{\beta}_{\mathbf{k}})$ in (4.5) can then be expressed as

$$l(\mathbf{y};\boldsymbol{\beta}_{\mathbf{k}}) = \boldsymbol{\delta}^T\big(\mathbf{X}_{\mathbf{k}}\boldsymbol{\beta}_{\mathbf{k}} - \log(\boldsymbol{\psi}_{\mathbf{k}})\big). \tag{4.15}$$

For each $n \times k$ design matrix $\mathbf{X}_{\mathbf{k}}$ and $\boldsymbol{\beta}_{\mathbf{k}}$ vector, define a new $k \times n$ matrix $\mathbb{X}_{\mathbf{k}}$, where its i^{th} column is obtained by

$$\mathbb{X}_{\mathbf{k}(i)} = \frac{(\overline{\mathbf{X}}_{\mathbf{k}i})(\overline{\boldsymbol{\eta}}_i)}{\psi_{\mathbf{k}_i}}. \tag{4.16}$$

Hhere, $\overline{\mathbf{X}}_{\mathbf{k}i}$ and $\overline{\boldsymbol{\eta}}_i$ are obtained as discussed before. This is repeated for every $1 \leq i \leq n$ to obtain the matrix $\mathbb{X}_{\mathbf{k}}$.

As a result, the negative gradient of $l(\mathbf{y};\boldsymbol{\beta}_{\mathbf{k}})$ cn be written as

$$-\frac{\partial l(\mathbf{y};\boldsymbol{\beta}_{\mathbf{k}})}{\partial \boldsymbol{\beta}_{\mathbf{k}}} = \big[\mathbb{X}_{\mathbf{k}} - \mathbf{X}_{\mathbf{k}}{}^T\big]\boldsymbol{\delta}. \tag{4.17}$$

To compute the Hessian matrix, let $\mathbb{X}_{\mathbf{k}_{ji}}$ be the element in row j and column i of $\mathbb{X}_{\mathbf{k}}$ matrix defined in (4.16). In what follows, the $k \times k$ identity matrix is denoted by $\mathbf{I}_k$ and $D(\boldsymbol{\alpha})$ denotes a diagonal matrix with the elements of the vector $\boldsymbol{\alpha}$ on its diagonal. Finally,

let $\boldsymbol{\zeta}^j = \mathbf{X}_{\mathbf{k}(j)}$, the j^{th} column of $\mathbf{X}_{\mathbf{k}}$.

The j^{th} row of the $k \times k$ Hessian matrix of $-l(\mathbf{y}; \boldsymbol{\beta}_{\mathbf{k}})$ is derived as

$$-\frac{\partial^2 l(\boldsymbol{\beta}_{\mathbf{k}})}{\partial \boldsymbol{\beta}_{\mathbf{k}j} \partial \boldsymbol{\beta}_{\mathbf{k}}{}^T} = \boldsymbol{\delta}^T_{1\times n} \boldsymbol{\Omega}^j_{n\times k}. \tag{4.18}$$

The $\boldsymbol{\Omega}^j_{n\times k}$ matrix itself is constructed row by row and its i^{th} row is computed by

$$\boldsymbol{\Omega}^j_i = \left[\overline{\mathbf{X}}_{\mathbf{k}i} \frac{D(\overline{\boldsymbol{\zeta}}^j_i)}{\psi_{\mathbf{k}_i}} \overline{\boldsymbol{\eta}}_i - \mathbb{X}_{\mathbf{k}ji} \mathbb{X}_{\mathbf{k}(i)}\right]^T. \tag{4.19}$$

The gradient and Hessian of the logarithm of piMOM prior is more straightforward. That is,

$$-\frac{\partial l_\pi(\boldsymbol{\beta}_{\mathbf{k}})}{\partial \boldsymbol{\beta}_{\mathbf{k}i}} = \frac{r+1}{\boldsymbol{\beta}_{\mathbf{k}i}} - \frac{2\tau}{\boldsymbol{\beta}_{\mathbf{k}i}{}^3}, \tag{4.20}$$

while the Hessian of $-l_\pi(\boldsymbol{\beta}_{\mathbf{k}})$ is a diagonal matrix, $D(\boldsymbol{\alpha})$, where

$$\alpha_i = \frac{6\tau}{\boldsymbol{\beta}_{\mathbf{k}i}{}^4} - \frac{r+1}{\boldsymbol{\beta}_{\mathbf{k}i}{}^2}. \tag{4.21}$$

Consequently, the gradient and Hessian matrix of $g(\boldsymbol{\beta}_{\mathbf{k}})$ is obtained using equations (4.17) to (4.21). These expressions are then used in finding the MAP, as well as computing the Laplace approximation to the marginal probability of $\mathbf{y}$.

I use limited memory version of Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS) for the optimization problem of finding the MAP. The initial value for the algorithm was $\hat{\boldsymbol{\beta}}_{\mathbf{k}}$, the MLE for the Cox proportional hazard model in (4.15).

Having all the components of formula (4.6), it is now possible to set up a MCMC framework to sample from the posterior distribution on the model space. The same technique of birth-death scheme, similar to that used in Nikooienejad et al. (2016), can be exploited here. However, computing the Hessian matrix in (4.19), which has complexity

of $O(n^3)$, makes MCMC iterations overwhelmingly slow and the whole process infeasible. An alternative stochastic search based approach to search the model space is discussed in the following section.

The estimated HPPM is defined as the highest posterior probability model among all visited models. In numerous applications many models are within a small margin of the HPPM. For this reason, it is tempting to obtain the Median Probability Model (MPM) (Barbieri et al., 2004), which is a model containing covariates that have posterior inclusion probability of at least 0.5. According to Barbieri et al. (2004), the posterior inclusion probability for covariate i is defined as

$$p_i = \sum_{\mathbf{k}:\ \mathbf{k}_i=1} p(M_{\mathbf{k}}|\mathbf{y}). \tag{4.22}$$

That is, the sum of posterior probabilities of all models that have covariate i as one of their variables. In this expression, $\mathbf{k}_i$ is a binary value determining the i^{th} covariate is included in model $\mathbf{k}$ or not.

4.2.3.2 Stochastic Search Algorithm

I utilize the S5 technique, proposed by Shin et al. (2015) for variable selection in linear regression problems, and adopt it for the survival data model. It is a stochastic search method that screens covariates at each step. The algorithm is scalable and its computational complexity is independent of p (Shin et al., 2015).

Screening is the essential part of the S5 algorithm. In linear regression, screening is defined based on the correlation between remaining covariates and the residuals of the regression using the current model (Fan and Lv, 2008). The concept of screening covariates for survival response data is proposed in Fan et al. (2010) and is defined based on the marginal utility for each covariate.

To illustrate the screening techniue, suppose the current model is $\mathbf{k}$. The conditional

utility of covariate $m \in \mathbf{k}^c$ is basically the amount of information it contributes to the survival outcome, given model $\mathbf{k}$, and is defined as

$$u_{m\,|\,\mathbf{k}} = \max_{\substack{\beta_m \\ m\in\mathbf{k}^c}} \boldsymbol{\delta}^T \Big[(\beta_m \mathbf{X}_{(m)} + \mathbf{X}_{\mathbf{k}} \boldsymbol{\beta}_{\mathbf{k}}) - \log \Big\{ \sum_{j=i}^{n} \exp(\beta_m x_{jm} + \mathbf{X}_{\mathbf{k}j} \boldsymbol{\beta}_{\mathbf{k}}) \Big\} \Big]. \tag{4.23}$$

By comparing this to (4.15), it can be seen that the conditional utility is the maximum likelihood for covariate m, while accounting for the information provided by model $\mathbf{k}$. Finding $u_{m\,|\,\mathbf{k}}$ is a univariate optimization procedure and thus fast to compute.

The S5 algorithm for surivial data works as follows. At each step, the d covariates with highest conditional utility are candidates to be added to the current model $\mathbf{k}$ and comprise the addition set, Γ^+. Hence, Γ^+ contains d models of size $k+1$ each. The deletion set, Γ^- comprises models with the same covariates as current model except with one variable that is removed. Therefore, Γ^- has k models of size $k-1$. From the current model, $\mathbf{k}$, I can potentially move to each of its neighbors in Γ^+ and Γ^-, with a probability proportional to the marginal probabilities of those neighboring models. This is how the model space is explored.

Note that this is done in a simulated annealing fashion and the marginal probabilities are raised to the power of $1/t_i$ where t_i is the i^{th} temperature in the annealing schedule and the temperatures decrease. To increase the number of visited models, a specified number of iterations are performed at each temperature. At the end, the model with the highest posterior probability out of those visited models is picked as the HPPM. More details of this technique are provided in Shin et al. (2015).

In my algorithm, I used 10 equally spaced temperatures varying from 3 to 1 and 30 iterations within each temperature. To increase the number of visitedmodels, I parallelize

the S5 procedure so that it could be distributed to multiple CPUs. Each CPU then executes S5 algorithm indepednently with a different starting model. All visited models are pooled together at the end and the HPPM is selected from all visited models. In my simulations, I found 120 CPUs were sufficient to explore the model space for design matrices with $O(10^4)$ covariates.

The proposed algorithm of Bayesian variable selection for survival data is implemented in the same R package with the methods for binary data discussed in Chapter 6.

4.3 Results

To investigate the performance of the proposed model selection procedure, I applied my method to both simulated data sets and real cancer genomic data. For simulation data, I compared the performance of my algorithm to ISIS-SCAD (Fan et al., 2010). The *SIS* package in R does not support the SCAD penalty for survival outcome and therefore for the real data, I compared my algorithm to a recently introduced method, named CoxHD, for classification of genes in Acute Myeloid Leukemia (AML). This algorithm has shown promising performacnce in selecting genes (Papaemmanuil et al., 2016). Although CoxHD is suitable for cases when $n \approx p$, it does not work well for $p \gg n$. Finally I applied my method to renal cell carcinoma reported in Cancer Genome Atlas Research Network (2013). We were not able to run the CoxHD algorithm for this dataset with $p \gg n$, given the runtime limits imposed by our High Performance Computational facility. Therefore, the results for renal cell carcinoma are only available for my method.

4.3.1 Simulation Studies

We first examined the six different simulation settings described in Fan et al. (2010). These settings consider different aspects of variable selection with respect to the correlation between true covariates and the magnitude of true coefficients. Here, I report two

of the hardest settings which are named as *Equi-correlated covariates with correlation = 0.5* and *two very hard variables to be selected.* I refer to these settings as case 1 and 2, respectively.

For case 1, $X_1, \ldots, X_p$ are multivariate Gaussian randon variable with mean 0 and marginal variance of 1. The correlation structure is $\text{corr}(X_i, X_j) = 0.5$ for $i \neq j$. The size of the true model is siz with true regression coefficients $\beta_1 = -1.5140, \beta_2 = 1.2799, \beta_3 = -1.5307, \beta_4 = 1.5164, \beta_5 = -1.3020, \beta_6 = 1.5833$ and $\beta_i = 0$ for all $i > 6$. The number of observations and covariates are $n = 400$ and $p = 1000$. The censoring rate for this simulation case is 23%.

For case 2, $X_1, \ldots, X_p$ are multivariate Gaussian random variables with mean 0 and marginal variance of 1. The correlation structure is $\text{corr}(X_i, X_5) = 0$ for all $i \neq 5$, $\text{corr}(X_i, X_4) = 1/\sqrt{2}$ for all $i \in \{4, 5\}$ and $\text{corr}(X_i, X_j) = 0.5$ for $i, j \in \{1, \ldots, p\} \setminus \{4, 5\},\ i \neq j$. The size of the true model is five with true regression coefficients $\beta_1 = 4, \beta_2 = 4, \beta_3 = 4, \beta_4 = -6\sqrt{2}, \beta_5 = 4/3$ and $\beta_i = 0$ for all $i > 5$. The corresponding censoring rate is 36% for this case. Similar to the previous case, The number of observations and covariates are $n = 400$ and $p = 1000$. The censoring rate for this simulation setting is 36%.

In both cases the survival times are simulated from an exponential distribution with mean 10, measuring that the baseline hazard function was $h_0(t) = 0.1$ for $t \geq 0$.

To measure the performance of the algorithms, I repeated each simulation setting 50 different times and at the end four different outcomes are reported. The first two of those outcomes are the median L_1 norm and the median squared L_2 norm for coefficient estimation error, denoted by ML_1 and ML_2 respectively. The L_1 norm is computed as $\sum_{i=1}^{p} |\hat{\beta}_i - \beta_i|$, where the squared L_2 norm is computed as $\sum_{i=1}^{p} (\hat{\beta}_i - \beta_i)^2$. Here, $\hat{\boldsymbol{\beta}}$ is the estimated and $\boldsymbol{\beta}$ is the true coefficient vector. The third outcome that is considered is the median model size of the selected models in 50 different iterations which is denoted

Table 4.1: Comparison between BVSNLP and ISIS-SCAD for simulation cases 1 and 2. $n = 400$ and $p = 1000$.

	BVSNLP	Van-ISIS	Var1-ISIS	Var2-ISIS
Case 1:				
ML_1	0.43	0.52	0.55	0.51
ML_2	0.04	0.07	0.08	0.07
MMS	6	6	6	6
P	1	1	1	1
Case 2:				
ML_1	0.77	0.99	1.1	1.29
ML_2	0.16	0.39	0.44	1.35
MMS	5	5	5	5
P	1	1	1	0.99

by MMS. The last out come is the proportion of times that the selected model contains all true variables. This parameter is denoted by P.

Table 4.1 shows the performance comparison between my method, BVSNLP and three different versions of ISIS-SCAD algorithm. The LASSO method is not in listed in that table because it takes several days to complete a single repetition of any of these simulation cases (Fan et al., 2010).

In the S5 algorithm, 30 iterations are used within each temperature. The parameter d was chosen as $2\lceil \log(p) \rceil$. Each S5 algorithm was run in parallel on 120 CPUs for both simultaion cases. The Beta binomial prior was used for the model space with average model size equal to 1. The hyperparameters were selected using the algorithm discussed in section 4.2.

As demonstrated in Table 4.1, my method performed better in estimating true coefficients compared to the best variants of the ISIS-SCAD algorithm. In addition, the BVSNLP algorithm chose the correct model with zero false positives in all 50 iterations for both simulation scenarios.

4.3.2 Real Data

I have studied two major datasets. The first dataset contains leukemia patients' survival times and was introduced in Papaemmanuil et al. (2016). For those data, the number of observations is almost the same as number of covariates ($n \approx p$). The other data set involves survival times for renal cell carcinoma in which $p \gg n$. This dataset was previously considered in Nikooienejad et al. (2016), where survival patients were converted to binary outcomes.

4.3.2.1 Leukemia Data

The recent work of Papaemmanuil et al. (2016) considers genomic classification in Acute Myeloid Leukemia (AML) patients. In this study, 1540 patients in three prospective trials were enrolled in order to investigate the effects of known mutated genes in AML. The censoring rate for this dataset is 41.3%.

Papaemmanuil et al. (2016) implemented a sparse random effects model for the Cox proportional hazards using their proposed R package, CoxHD. In that method, the parameters of random effects model are assumed to come from a shared Normal distribution. The coefficients are then obtained by finding the maximum a posteriori (MAP) of a penalized partial likelihood function. The covariates are clustered into different groups with a shared mean and variance. The shared mean vector and covariance matrix are estimated iteratively using the expectation Maximization (EM) algorithm. This method has shown to have a dominant predictive performance compared to existing frequentist methods for datasets where $n \approx p$ and henceforth is denoted by NEJM.

To compare NEJM method to ours, I used the exact same dataset they provided when computing the predictive accuracy of the CoxHD method. That dataset contains 229 covariates in 8 different categories. Those categories are listed as follows, with the number of covariates at each category listed in the parenthesis. Clinical (11), copy number alter-

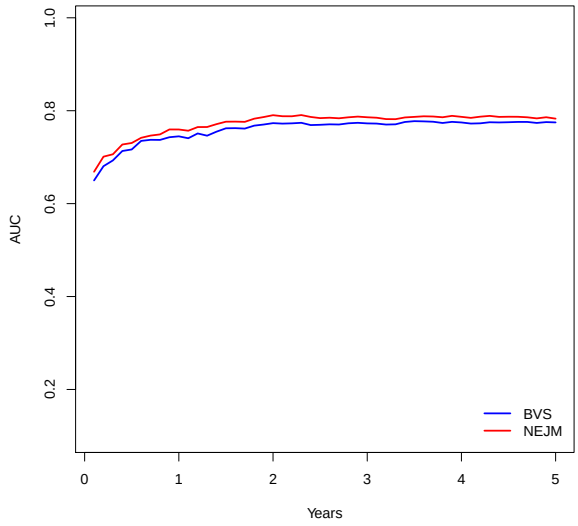


Figure 4.1: Average AUC of both BVSNLP and CoxHD methods after 5 fold cross validation for AML dataset.

ations (18), demographics (2), fusion genes (8), genes (58), gene:gene interactions (126), nuisances (4) and treatment (2). In this list, nuisance variables are variables such as the trial a patient was enrolled in, the year a patient entered the clinical trial and whether cytogenetic data were missing. Moreover, gene:gene interactions are between two mutated genes.

A five fold cross validation was performed to measure the predictive accuracy of both algorithms. The Area Under Curve (AUC) for right censored data is used to evaluate the predictive power for each of the methods. To estimate AUC, I exploited Uno's method (Uno et al., 2007), available in an R package named *survAUC* (Potapov et al., 2012).

Figure 4.1 illustrates the predictive power for both methods. The BVSNLP method performs similarly to the NEJM method. The mean square difference between predictive AUC curves is only 1.9×10^{-4}.

4.3.2.2 Renal Cell Carcinoma Data

This dataset was generated by Cancer Genome Atlas Research Network (2013) and contains Illumina HiSeq data on mRNA expression for 467 patient samples. The survival outcomes of these patients were available. A preprocessing step using DeMix algorithm (Ahn et al., 2013) was performed on the data in order to remove stromal contamination. The resulting number of observations included in my analysis was 193, with 14,149 gene expression covariates in the design matrix. The censoring rate for this dataset is 60.6%.

I applied my method, BVSNLP to this dataset for two purposes. I wanted to find significant genes and evaluate the predictive accuracy of my method. TI again performed a five fold cross validation and measured AUC for each fold, averaging them at the end. The *survAUC* package was used to compute predictive AUC. Figure 4.2 shows the average predictive AUC.

The only gene that is appearing in my algorithm's HPPM was CDC7. This is an important gene in the cell division cycle and DNA replication and belongs to the cell cycle pathway. CDC7 has gained some attention as a potential pathway for cancer treatment (Montagnoli et al., 2010). I also examined the posterior inclusion probabilities for all genes. The 5 genes with the highest posterior inclusion probabilities were CDC7, NUMBL, CNTNAP1, CCNF and ADAMTS14, with posterior inclusion probabilities of $0.213, 0.149, 0.083, 0.066$ and 0.049, respectively. The NUMBL gene was also reported in the selected genes of my previous analysis in Nikooienejad et al. (2016), which was discussed in section 3.4.1.

4.4 Discussion

In this chapter a Bayesian method, named BVSNLP, for selecting variables in high and ultrahigh dimensional datasets with survival time otcomes was proposed. My method imposes inverse moment nonlocal prior density on non-zero regression coefficients. This

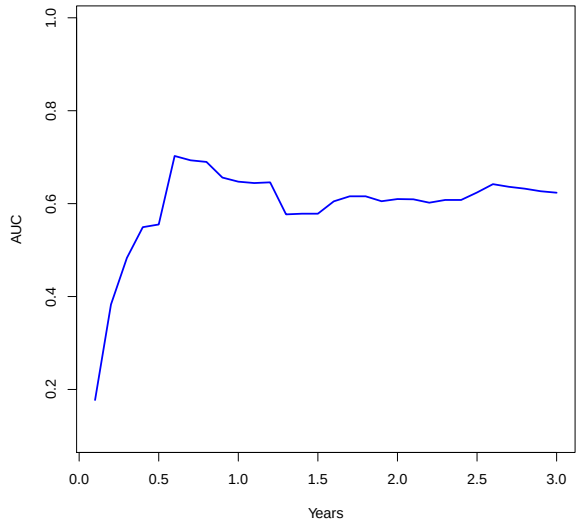


Figure 4.2: Average AUC of BVSNLP method after 5 fold cross validation for renal cell carcinoma dataset.

positively impacts variable selection and coefficient estimation performance, as demonstrated by simulation studies.

Two real datasets were considered in this chapter. BVSNLP found sparse models with biologically relevant genes that comply with previous findings in both cases. The proposed method showed a reliable predictive accuracy as measured by predictive AUC and outperformed competing methods. It should be noted that my algorithm complexity increases with the sample size at an $O(n^3)$ rate, which slows down the processing of datasets with thousands of observations.

My algorithm is implemented in an R package which is discribed in Chapter 6.

5. ON EXISTENCE AND DERIVATION OF UNIFORMLY MOST POWERFUL BAYESIAN TESTS WITH APPLICATION TO NON-CENTRAL χ^2 TESTS

5.1 Introduction

Bayesian hypothesis tests are based on computing the posterior probabilities of competing hypotheses given data. From Bayes theorem, the posterior probability of each hypothesis is proportional to the product of its prior probability and the marginal likelihood of the data given that the hypothesis is true. In the case of two competing hypotheses, the posterior odds between hypotheses H_0 and H_1 can be written as

$$\frac{\mathbf{P}(H_1 \,|\, \mathbf{x})}{\mathbf{P}(H_0 \,|\, \mathbf{x})} = \frac{m_1(\mathbf{x})}{m_0(\mathbf{x})} \times \frac{p(H_1)}{p(H_0)}, \tag{5.1}$$

where $m_1(\mathbf{x})/m_0(\mathbf{x})$ is called *Bayes factor* in favor of the alternative hypothesis (denoted more simply as $\text{BF}_{10}(\mathbf{x})$), $m_i(\mathbf{x})$ denotes the marginal density of the data under hypothesis i, and $p(H_i)$ denotes the prior probability of hypothesis H_i. The logarithm of the Bayes factor is called the *weight of evidence*. I assume throughout that the sampling density of the data $\mathbf{x}$ is defined with respect to a $\sigma-$finite measure and is described by the same parametric family of densities indexed by a parameter $\theta \in \mathbb{R}$ under all hypotheses, and refer to models and hypotheses interchangeably. Letting $f(\mathbf{x} \,|\, \theta)$ denote the sampling density of the data $\mathbf{x}$ given the value of a parameter $\theta \in \Theta$, and $\pi_i(\theta)$ the prior on θ given hypothesis i, the marginal density of the data under hypothesis i can be written as

$$m_i(\mathbf{x}) = \int_\Theta f(\mathbf{x} \,|\, \theta)\pi_i(\theta)d\theta.$$

In the classical testing paradigm, a decision to reject the null hypothesis (denoted throughout this chapter as H_0) occurs when the value of a test statistic exceeds a spec-

ified threshold. UMPBTs are defined in a similar way by assuming that H_0 is rejected in favor of an alternative hypothesis H_1 if $\text{BF}_{10}(\mathbf{x})$, exceeds a pre-specified threshold, say γ.

With this notation and assumptions, a UMPBT(γ) was defined in Johnson (2013c) as follows:

Definition 5.1.1. A uniformly most powerful Bayesian test for evidence threshold $\gamma > 0$ in favor of the alternative hypothesis H_1 against a fixed null hypothesis H_0, denoted by UMPBT(γ), is a Bayesian hypothesis test in which the Bayes factor for the test satisfies the following inequality for any $\theta_t \in \Theta$ and for all alternative hypotheses $H_2 : \theta \sim \pi_2(\theta)$:

$$\mathbf{P}_{\theta_t}\big[\text{BF}_{10}(\mathbf{x}) > \gamma\big] \geq \mathbf{P}_{\theta_t}\big[\text{BF}_{20}(\mathbf{x}) > \gamma\big]. \tag{5.2}$$

The alternative hypothesis H_1 in (5.2) maximizes the probability that the Bayes factor is greater than a fixed evidence threshold, γ, among all possible alternatives and for all possible values of the data-generating parameter θ_t.

For the case of testing simple null hypotheses $H_0 : \theta = \theta_0$, and under the further assumption that tests are one-sided (i.e., $\Theta = \{\theta : \theta > \theta_0\}$ or $\Theta = \{\theta : \theta < \theta_0\}$), UMPBTs for one parameter exponential families were derived in Johnson (2013c). These tests included tests of binomial proportion, tests of normal means with known variance, tests for normal variances when the mean is known, and tests that the non-centrality parameter of χ^2_1 distribution is equal to zero (Johnson, 2013b,c). UMPBTs were extended in Goddard and Johnson (2016) by restricting the class of alternative hypotheses over which the maximization in (5.2) is performed.

The UMPBTs derived in (Johnson, 2013c) were all obtained by rewriting $\mathbf{P}_{\theta_t}[\text{BF}_{10}(\mathbf{x}) > \gamma]$ in (5.2) as

$$\mathbf{P}_{\theta_t}\big[h(\mathbf{x}) > A(\gamma, \theta)\big]. \tag{5.3}$$

where $h(\mathbf{x})$ is a function of the data. By so doing, the probability in (5.3) can be maximized

with respect to θ by simply minimizing $A(\gamma, \theta)$, regardless of the distribution of $h(\mathbf{x})$, thus producing a UMPBT(γ) test.

The main motivation behind this chapter is to provide a new approach to defining UMPBTs when rewriting $\mathbf{P}_{\theta_t}[\text{BF}_{10}(\mathbf{x}) > \gamma]$ as (5.3) cannot be achieved. A primary application of this general method is to derive UMPBTs for tests of non-centrality parameters in χ^2 distributions with arbitrary degrees of freedom.

The remainder of this chapter is organized as follows. Section 5.2 discusses generalization of methodology to determine the existence of UMPBTs. In Section 5.3 I exploit the new methodology to derive the UMPBT(γ) of a non-centrality parameter of a χ^2_ν distribution with $\nu > 1$ degrees of freedom. This test is important for tests of independence in contingency tables, in likelihood ratio and score tests. Several diagnostic plots are provided in Section 5.4. Concluding comments appear in Section 5.5.

5.2 Method

5.2.1 Preliminaries

Let $y = h(\mathbf{x})$ denote a sufficient statistic of the data, with $y \in \mathbb{R}$,For ease of notation, I suppress dependence on $\mathbf{x}$ and write $\text{BF}_{10} = \text{BF}_{10}(\mathbf{x})$. I also restrict attention to simple null hypotheses $\theta_0 \in \Theta_0$. For every simple alternative $\theta_1 \in \Theta_1$, I denote the Bayes Factor in favor of θ_1 as $g(y, \theta_1)$.

Let $\Omega_\gamma(\theta_1) \subset \mathbb{R}$ denote the regions where $g(y, \theta_1) > \gamma$. That is, $\Omega_\gamma(\theta_1)$ represents the rejection region when the null hypothesis is rejected in favor of the alternative $H_1 : \theta = \theta_1$ with respect to a fixed threshold γ. Specifically,

$$\Omega_\gamma(\theta_1) = \{y : g(y, \theta_1) > \gamma\}. \tag{5.4}$$

Let $f(y; \theta_t)$ be the density function of y for the true data generating parameter, θ_t, and

F its corresponding distribution function defined with respect to a σ-finite measure, μ. Also let $S(f) \subset \mathbb{R}$ denote the support of f. Define a and b as

$$a(\theta_t) = \inf S(f) \qquad b(\theta_t) = \sup S(f). \tag{5.5}$$

Next, define $H_\gamma(\theta_1; \theta_t) \geq 0$ to be

$$H_\gamma(\theta_1; \theta_t) = \mathbf{P}_{\theta_t}[g(y, \theta) > \gamma] = \int_{\Omega_\gamma(\theta_1)} F(y; \theta_t) d\mu, \tag{5.6}$$

the probability that the null hypothesis is rejected when the true state of nature is θ_t and the alternative is specified as $H_1 : \theta = \theta_1$.

If θ^* satisfies

$$\theta^* = \underset{\theta \in \Theta_1}{\operatorname{argmax}} H_\gamma(\theta_1; \theta_t) \quad \forall \theta_t \in \Theta, \tag{5.7}$$

then it follows that $H_1 : \theta = \theta^*$ defines the UMPBT(γ).

5.2.2 Existence and Derivation of UMPBT

Using the notation described above, I next describe a sufficient condition for the existence of a UMPBT in a one-sided test of hypotheses.

Theorem 5.2.1. *For a Bayesian test of hypotheses with a point null $H_0 : \theta = \theta_0$ against one sided alternative hypothesis and a fixed threshold γ, $\theta^* \in \Theta_1$ is the parameter value that defines the alternative hypothesis corresponding to a UMPBT(γ) if*

$$\Omega_\gamma(\theta_1) \subset \Omega_\gamma(\theta^*); \quad \textit{for all } \theta \in \Theta_1 \textit{ and } \theta \neq \theta^*. \tag{5.8}$$

That is, the rejection region of θ^* covers the rejection region that is generated under all alternative parameters.

Proof: Given the relation in (5.8), following the definition of the function $H_\gamma(\theta_1;\theta_t)$ in (5.6),

$$H_\gamma(\theta_1;\theta_t) = \int_{\Omega_\gamma(\theta_1)} F(dy;\theta_t) < \int_{\Omega_\gamma(\theta_1^*)} F(dy;\theta_t) = H_\gamma(\theta^*;\theta_t). \tag{5.9}$$

Knowing that $\theta^* \in \Theta_1$, the inequality above ensures that $\theta^* = \text{argmax}_{\theta\in\Theta_1} H_\gamma(\theta_1;\theta_t)$ and the proof is complete. □

This is a useful existence theorem for UMPBTs. For a special case of Theorem 5.2.1 when the Bayes factor is a continuous and differentiable function of y, a more practical mechanism for establishing a sufficient condition for the existence of a UMPBT can be achieved. This condition is provided in the following corollary.

Corollary 5.2.2. *For a Bayesian test of hypotheses with a point null $H_0 : \theta = \theta_0$ against one sided alternative hypothesis, let the Bayes factor, BF_{10}, be a continuous differentiable function in the domain of y, for every alternative parameter $\theta \in \Theta_1$. For a fixed threshold γ, the UMPBT(γ), exists if the rejection region defined in (5.4) is either of the form of $\big(a(\theta_t), y^*(\theta)\big)$ or $\big(y^*(\theta), b(\theta_t)\big)$ for all θ_t and $\theta \in \Theta_1$. Parameters $a(\theta_t)$ and $b(\theta_t)$ are defined in (5.5). The value θ that provides the alternative hypothesis for the UMPBT(γ) is defined as:*

$$\theta^* = \underset{\theta}{\text{argmin}}\, v y^*(\theta) \;\; \textit{where} \;\; v = \begin{cases} 1 & \textit{if } \Omega_\gamma(\theta_1) = \big(y^*(\theta), b(\theta_t)\big) \\ -1 & \textit{if } \Omega_\gamma(\theta_1) = \big(a(\theta_t), y^*(\theta)\big) \end{cases} \tag{5.10}$$

Proof: To show that form of rejection region defined in (5.10) results in existence of a

UMPBT(γ), notice that $H_\gamma(\theta_1;\theta_t)$ can be expressed as,

$$H_\gamma(\theta_1;\theta_t) = \begin{cases} \int_{y^*(\theta)}^{b(\theta_t)} F(dy;\theta_t) = 1 - F\big(y^*(\theta);\theta_t\big); & v = 1 \\ \int_{a(\theta_t)}^{y^*(\theta)} F(dy;\theta_t) = F\big(y^*(\theta);\theta_t\big); & v = -1 \end{cases}.$$

For each θ_t, this implies that $H_\gamma(\theta_1;\theta_t)$ is maximized whenever $vy^*(\theta)$ is minimized. Minimizing $vy^*(\theta)$ does not depend on the true parameter of the distribution and it can be found regardless of θ_t. That means θ^*, the θ that produces the smallest $vy^*(\theta)$, is constant for every θ_t and every $\gamma > 1$. The value of θ^* is thus equal to the alternative parameter corresponding to the UMPBT(γ).

Corollary 5.2.2 offers a simple tool to check the existence of UMPBT for continuous distributions, as well as offering a practical approach for finding it. A potential first step to use Corollary 5.2.2 is to identify the rejection region by determining the values of y that satisfy

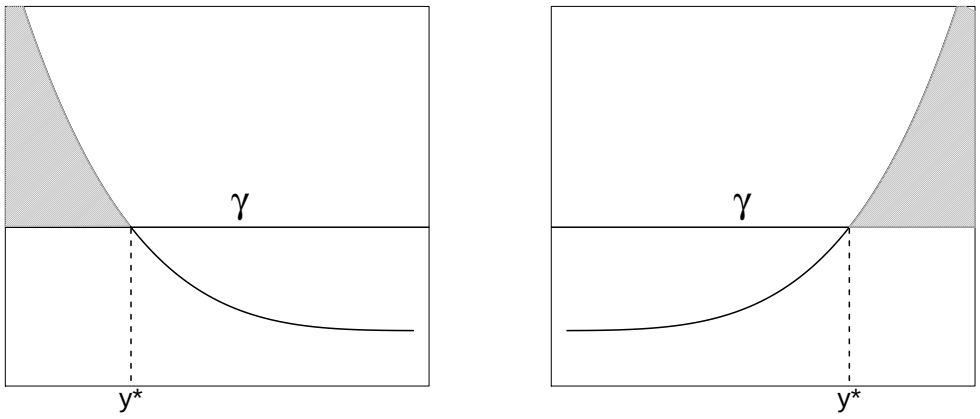
$$g(y,\theta) - \gamma = 0. \tag{5.11}$$

The following Theorem provides even more practical way of exploiting Corollary 5.2.2.

Theorem 5.2.3. *Let $Q(\theta;y) = \frac{\partial g(y,\theta)}{\partial y}$ be the first derivative of BF_{10} with respect to y. Suppose that for all θ and for all y, $Q(\theta;y) > 0$ or $Q(\theta;y) < 0$ and let v denote the sign of $Q(\theta;y)$. Then UMPBT(γ) exists, and θ^*, the parameter that defines the UMPBT(γ) alternative hypothesis, satisfies*

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, vy_0(\theta);\ \textit{where}\ g(y_0,\lambda) - \gamma = 0. \tag{5.12}$$

Proof: If Q is strictly positive or negative, then the function g is a one to one function



(a) $v = -1$, $\Omega_\gamma(\theta_1) = (-\infty, y^*(\theta))$ (b) $v = 1$ $\Omega_\gamma(\theta_1) = (y^*(\theta), \infty)$

Figure 5.1: Relation between increasing or decreasing nature of the Bayes factor and the type of boundedness in $\Omega_\gamma(\theta_1)$.

and hence $g(y, \theta) - \gamma$ has only one unique root, say $y^*(\theta)$. Having a unique $y^*(\theta)$, the rejection region is either the region on the right of the root, $\big(y^*(\theta), b(\theta_t)\big)$ or on its left, $\big(a(\theta_t), y^*(\theta)\big)$. The form of the rejection region, $\Omega_\gamma(\theta_1)$, depends on v. More specifically, $\Omega_\gamma(\theta_1)$ is of the form $\big(y^*(\theta), b(\theta_t)\big)$ when $v = 1$ and it is of the form $\big(a(\theta_t), y^*(\theta)\big)$ when $uv = -1$. This fact is illustrated in Figure 5.1 when $a = -\infty$ and $b = +\infty$. Using Corollary 5.2.2, the statement in the corollary follows and the proof is complete. □

Theorem 5.2.3 provides a special case of Corollary 5.2.2 when the Bayes factor is a *monotone* function.

5.3 UMPBTs for Common Tests of Hypotheses

Theorems 5.2.1, 5.2.3 and Corollary 5.2.2 introduced a general platform for existence and derivation of UMPBTs. In this section, these theorems are used to find UMPBTs for common hypothesis tests.

5.3.1 UMPBT for Chi-squared Tests

Let x be an observation from a chi-squared distribution on ν degrees of freedom and non-centrality parameter λ, denoted by $\chi^2_\nu(\lambda)$ distribution. As shown in Patnaik (1949) and Seber (1963), the probability density function of a $\chi^2_\nu(\lambda)$ random variable can be written as

$$f(x \mid \lambda) = \frac{1}{2} \exp^{-(x+\lambda)/2} \left(\frac{x}{\lambda}\right)^{\nu/4-1/2} I_{\nu/2-1}(\sqrt{\lambda x}). \tag{5.13}$$

Here, $I_\nu(y)$ is the modified Bessel function of the first kind and for a real valued ν is defined as

$$I_\nu(x) = \sum_{j=0}^{\infty} \frac{(x/2)^{2j+\nu}}{\Gamma(\nu + j + 1)j!}. \tag{5.14}$$

In general, the range of modified Bessel function of the first kind is $\mathbb{C}$, the set of all complex numbers. However, for real positive arguments and real-valued degrees of freedom, the range is $\mathbb{R}^+$. In the case of $\lambda = 0$, the probability distribution function in (5.13) reduces to

$$f(x|\lambda = 0) = (\frac{1}{2})^{\nu/2} \exp^{-x/2} \frac{x^{\nu/2-1}}{\Gamma(\nu/2)}. \tag{5.15}$$

We are concerned with testing $H_0 : \lambda = 0$ against $H_1 : \lambda > 0$. Using (5.13) and (5.15), the Bayes factor in favor of the alternative hypothesis can be expressed as

$$g(x, \lambda) = \Gamma(\frac{\nu}{2}) \exp^{-\lambda/2} 2^{\nu/2-1} (\sqrt{\lambda x})^{1-\nu/2} I_{\nu/2-1}(\sqrt{\lambda x}). \tag{5.16}$$

Here, the parameter y in Corollary 5.2.2 is the observed data x. For this Bayes factor, both the data and the parameter of the test are arguments of the modified Bessel function. Thus the rejection region can not be written in the form of (5.3). The following theorem proves the existence of UMPBT(γ) for this test using Corollary 5.2.2.

Theorem 5.3.1. *Suppose $x \sim \chi^2_\nu(\lambda)$ and consider the test of $H_0 : \lambda = 0$ versus $H_1 : \lambda >$*

0. *Given an evidence threshold $\gamma > 0$, a UMPBT(γ) exists, and the alternative hypothesis for this test is given by*

$$\theta^* = \underset{\lambda \in \Lambda_1}{\operatorname{argmin}}\, y^*(\lambda); \quad y^*(\lambda) \textit{ is the root of } g(y, \lambda) - \gamma. \tag{5.17}$$

Proof: The first derivative of the modified Bessel function of the first kind with ν degrees of freedom can be expressed as $\frac{\partial I_\nu(z)}{\partial z} = \frac{\nu}{z} I_\nu(z) + I_{\nu+1}(z)$. The first derivative of $g(y, \lambda)$ with respect to y is then equal to

$$\frac{\partial g(y, \lambda)}{\partial y} = \frac{\alpha}{2} \lambda (\sqrt{\lambda y})^{-\nu/2} I_{\nu/2}(\sqrt{\lambda y}), \tag{5.18}$$

where $\alpha = \Gamma(\frac{\nu}{2}) \exp^{-\lambda/2} 2^{\nu/2-1}$ and a positive number. The domain for the alternative hypothesis is $\Lambda_1 : \lambda > 0$ and the support of the Chi-squared distribution is $\mathbb{R}^+$ which results in a real positive modified Bessel function of the first kind. Therefore, the derivative in (5.18) is strictly positive. The result in the theorem then follows, using Theorem 5.2.3. Notice that in this test $v = 1$. □

5.3.2 Exponential Family Distributions

In one parameter tests of hypotheses for exponential family distributions with a point null, $H_0 : \theta = \theta_0$, against one sided arbitrary alternative hypotheses, let $\mathbf{x}$ be n i.i.d observations, $\{x_1, x_2, \cdots, x_n\}$, from one of the distributions in the exponential family. That is, the probability density function for each observation can be expressed as

$$f(x \,|\, \theta) = h(x) \exp[\eta(\theta) T(x) - A(\theta)], \tag{5.19}$$

where $h(x), A(\theta)$ and $\eta(\theta)$ are known functions and $T(x)$ is the sufficient statistic of the data. For n independent observations $\mathbf{x}$, the Bayes factor in favor of the alternative for the

test described above, can be expressed as

$$\text{BF}_{10} = \exp\big[n\big(A(\theta_0) - A(\theta)\big)\big] \exp\Big[\sum_{i=1}^{n} T(x_i)\big(\eta(\theta) - \eta(\theta_0)\big)\Big], \tag{5.20}$$

where $\theta \in \Theta_1$ is the parameter under the alternative hypothesis. In this formulation, sufficient statistic y in Corollary 5.2.2 is $y = \sum_{i=1}^{n} T(x_i)$. Consequently, the first derivative of the Bayes factor with respect to y in (5.20) can be specified by

$$\frac{\partial g(y,\theta)}{\partial y} = [\eta(\theta) - \eta(\theta_0)] \exp\big[n\big(A(\theta) - A(\theta_0)\big) + y\big(\eta(\theta) - \eta(\theta_0)\big)\big]. \tag{5.21}$$

If the function $\eta(\theta)$ is monotonic on Θ_1, the derivative above does not change sign and is strictly positive or negative. Therefore, for a fixed threshold γ, the function $q(y)$ in equation (5.11) has a unique root which is given by

$$y = \frac{\log(\gamma) + n\big(A(\theta) - A(\theta_0)\big)}{\eta(\theta) - \eta(\theta_0)}. \tag{5.22}$$

Following Theorem 5.2.3, θ^*, the alternative parameter corresponding to UMPBT(γ) exists and is derived by,

$$\theta^* = \underset{\theta \in \Theta_1}{\operatorname{argmin}}\, v \frac{\log(\gamma) + n\big(A(\theta) - A(\theta_0)\big)}{\eta(\theta) - \eta(\theta_0)}. \tag{5.23}$$

where v is defined as in Theorem 5.2.3, and in this case is equal to the sign of $\eta(\theta) - \eta(\theta_0)$, according to (5.22).

Accordingly, in testing one sided alternative against a point null hypothesis for one dimensional exponential family distributions, the UMPBT(γ) can always be found as described in (5.23), *only if* the natural parameter of the exponential family, $\eta(\theta)$ is monotone on the domain of the alternative hypothesis, Θ_1.

The above results and the formula (5.23) complies with the findings in Johnson (2013c). Notice that the value of v is determined by the monotonicity of $\eta(\theta)$ in Θ_1 and the direction of comparison in the alternative hypothesis. In general, the type of monotonicity in the Bayes factor depends on the alternative hypothesis.

5.4 Results

5.4.1 Analysis of Evidence Threshold

By using UMPBT(γ) to set the parameter in the alternative hypothesis, we can match the rejection region of classical tests of hypotheses to the Bayesian test. This allows us to compare the evidence threshold and p-values of the two tests. In this section, I investigate how the evidence threshold changes with respect to the degrees of freedom for a fixed size classical chi-squared test.

Figure 5.2 demonstrates this behavior. It is interesting that the threshold remains almost constant as the degrees of freedom increases. Thus the UMPBT provides a good insight on selecting evidence threshold before doing analysis by selecting the one equivalent to the required significance level in a classical test. This concept is illustrated in the following examples.

5.4.2 Test of Independence in Contingency Tables

Test of independence between rows and columns of contingency tables is a common test in standard statistical practice where the null hypothesis assumes rows and columns are independent. Performing this test in the Bayesian paradigm requires computation of the Bayes factor, which depends on prior densities for the multinomial probability vector under both hypotheses. Different methods have been proposed to define the aforementioned prior. Albert (1990) uses a prior distribution for the alternative constructed about the "independence surface", that is the null hypothesis. Good and Crook (1987) used a mixed-Dirichlet prior and checked the robustness and sensitivity with respect to hyperpri-

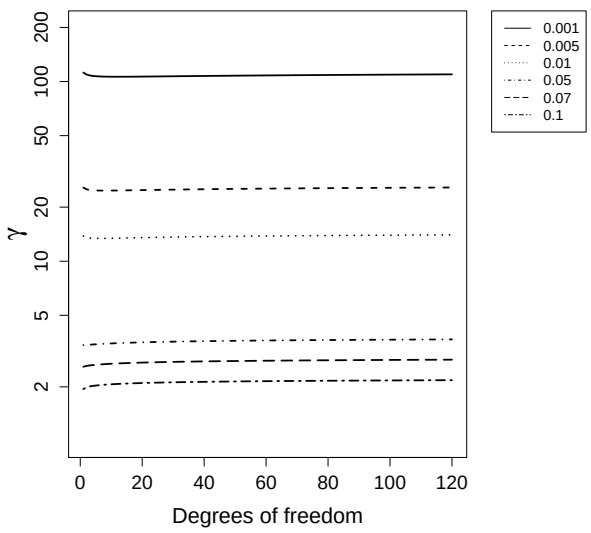


Figure 5.2: Evidence threshold vs. degrees of freedom in Chi-squared tests for different significance levels

ors and their hyperparameters. Johnson (2005) proposed a totally different approach by computing the Bayes factor based on a test statistic, in this case a χ^2-statistic.

My proposed method extends the concept of uniformly most powerful Bayesian tests to non-central χ^2 tests with different degrees of freedom. As a result, burrowing the methodology from Johnson (2005), I use a χ^2-statistic to compute the Bayes factor. The difference here is that the prior used for the non-centrality parameter is derived from a UMPBT. Johnson (2005) used the conjugate prior density, a gamma distribution for λ, to set this prior. The following example contrasts the performance of these methods.

The contingency table shown in Table 5.1 represents the cross classification on cancer site and blood type for patients with stomach cancer (White and Eisenberg, 1959). The total sample size is 707 and the goal is to test independence of rows and columns.

The χ^2-statistic for this contingency table is 12.65 on 6 degrees of freedom. Johnson (2005) computes the Bayes factor against the independence model as 2.97, when the pa-

Table 5.1: White and Eisenberg (1959) classification of cancer patients

Site	Results for the following blood groups:		
	O	A	B or AB
Pylorus and antrum	104	140	52
Body and fundus	116	117	52
Cardia	28	39	11
Extensive	28	12	8

rameter α in the proposed Bayes factor is chosen to maximize the marginal density of the data under the alternative hypothesis.

Following the recommendations for hyperparameters in Albert (1990), the maximum Bayes factor against the null hypothesis obtained by their model is 3.02. This is obtained by maximizing the approximate Bayes factor with respect to the parameter that controls the dispersion of the alternative around the independence surface. Under the model proposed by Good and Crook (1987), the Bayes factor is 3.06.

Using the methodology proposed in this chapter, the Bayes factors based on χ^2-statistic with non-centrality parameter corresponding to the UMPBT(γ) for the alternative hypothesis can be calculated for different evidence threshold values, γ. The Baye factor obtained from the UMPBTs associated with the thresholds for significance levels of 0.05, 0.01 and 0.005 on 6 degrees of freedom and depicted in Figure (5.2) and are 3.46, 13.40 and 24.74, respectively. The Bayes factors for this problem, calculated for those evidence thresholds are summarized in Table 5.2.

Table 5.2: Bayes factors based on χ^2-statistic and UMPBT(γ) non-centrality parameter for different threshold values

Significance Level	0.05	0.01	0.005
Equiv. Evidence Threshold	3.46	13.40	24.74
Bayes Factor against Null	3.52	2.93	2.50

As expected, the evidence threshold increases by decrease in the significance level. The Bayes factors are obtained by equation (5.16), where $x = 12.65$, $\nu = 6$ and λ is obtained for each evidence threshold using UMPBT. Since the χ^2 statistic is fixed, the Bayes factor decreases for more significant tests. This is because the value of the non-centrality parameter is increasing with the significance of the test and thus more evidence in the observations (more extreme values of χ^2 statistic) is needed to have a greater Bayes factor.

It is compelling to compare the values of the Bayes factors in Table 5.2 with the findings based on othe prior assumptions. By It is inferred that the priors that are used in those three methods results in tests that have equivalent significant levels between 0.05 and 0.01 in classical hypothesis testing.

5.5 Discussion

The methodology proposed in Johnson (2005) computes Bayes factors based on tests statistics to bypass the neccessity of defining subjective priors and the burden of computing marginal probabilities. However, it still requires a prior for the non-centrality parameter that is used in the test statistic under alternative. Johnson (2005) suggested that a conjugate or any other convenient prior be used for this purpose. The method in this chapter provides an alternative solution using UMPBTs based on χ^2 statistic.

The use of UMPBTs depends on the value of evidence threshold, γ. Finding a reasonable value for γ mainly depends on the required significance of the test as well as any other data specific knowledge on the problem. One can consult Kass and Raftery (1995) for more information on the value of evidence threshold for different levels of significance. Another way to set the evidence threshold is to match the rejection region of UMPBT with the one used in a classical hypothesis test and choosing the threshold that produces a rejection regions matched to a specific test size.

Theorem 5.2.3 can be contrasted to the Karlin-Rubin theorem (Karlin and Rubin, 1956). That theorem states that for a monotone non-decreasing likelihood ratio, in testing $H_0 : \theta = \theta_0$ against $H_1 : \theta > \theta_0$, the threshold test,

$$\phi(x) = \begin{cases} 1 & \text{if } x > x_0 \\ \\ 0 & \text{if } x < x_0 \end{cases}. \tag{5.24}$$

is uniformly most powerful test. I also showed that for a monotone Bayes factor, UMPBT always exists.

6. BVSNLP: THE R PACKAGE FOR HIGH DIMENSIONAL BAYESIAN VARIABLE SELECTION

6.1 Introduction

There exist myriad variable selection algorithms in both frequentist and Bayesian paradigms as discussed in chapter 1. There are, consequently, various R software packages that have been introduced for implementation of those algorithms. Some examples of the most common variable selection packages that exploit penalized likelihood methods are *SIS* (Fan et al., 2015) that exploits ISIS-SCAD algorithm, *glmnet* (Friedman et al., 2010) for LASSO and Elastic-net regularized generalized linear models, *lars* (Hastie and Efron, 2013) that is implemented for least angle regression algorithm for efficiently fitting lasso and *flare* (Li et al., 2015) for family of LASSO regression.

On the other hand, the number of Bayesian variable selection packages is not so large. The main reason for this is the intensive computational load of Bayesian methods. In recent years, however, an enormous increase of computational power and emerging of clusters to facilitate parallel computing, Bayesian methods have gained more attention and some R packages for high dimensional variable selection have been introduced. The package *mombf* (Rossell et al., 2015) was proposed for high dimensional variable selection in linear regression models using nonlocal priors. Along similar lines, *BayesS5* employs a stochastic search method with screening to find the highest posterior probability model. The package *BayesVarSel* (Garcia-Donato and Forte, 2016) was introduced for variable selection in linear models, while the *pogit* package (Dvorzak and Wagner, 2016) is implemented for Poisson-Logistic models. Another package in this class is *spikeSlabGAM* (Scheipl, 2011), which utilizes spike and slab priors for variable selection for generalized additive models.

The listed packages for Bayesian variable selection are mostly suited for linear regression models and there are none for high dimensional selection of logistic and survival data. In this chapter, I introduce a new R package that I have developed for such datasets based on the methodologies described in previous chapters.

This chapter is organized as follows. In section 6.2 I introduce the package and discuss the general points about its structure. Section 6.3 investigates important functions of the package by investigating the input and output arguments. Section 6.4 concludes the chapter.

6.2 General Points of BVSNLP Package

My proposed package is named BVSNLP, which stands for Bayesian Variable Selection Non Local Prior, reflecting that nonlocal priors are used for model coefficients. This package is essentially designed for high dimensional variable selection for datasets where the response vector is binary or survival times. I use logistic regression to model the outcome for the former whereas in the latter Cox proportional hazard models are used. For more details on the methods refer to chapters 2 and 3.

The main feature of the BVSNLP package is the use of the C++ language with object-oriented programming to speed up the process. This package supports parallel computing in order to perform the coupling algorithm in logistic regression variable selection (section 3.3.1) and parallel stochastic search algorithm in survival data analysis (section 4.2.3.2). In the latter, each CPU is responsible for one S5 algorithm and in the end all visited models from each S5 run are pooled together. The highest posterior probability model is chosen from this set.

To integrate R and C++, the package *Rcpp* (Eddelbuettel and François, 2011) is used. This package facilitates variable passing between the R environment and C++. For complex linear algebra computations, *RcppArmadillo* (Eddelbuettel and Sanderson, 2014) and

RcppEigen (Bates and Eddelbuettel, 2013) packages are employed. There are different instances of nonlinear optimization in my algorithm as described in previous chapters. Among different methods of nonlinear optimization, I found the limited memory version of Broyden-Fletcher-Goldfarb-Shanno algorithm, also known as L-BFGS, to be robust as well as efficient for the functions. For a good review of different optimization methods one can consult Mullen (2014). To perform this optimization algorithm, I use *RcppNumerical* package (Qiu et al., 2016) which is based on the Eigen, a template C++ library for matrices, vectors, numerical solvers, etc. (Guennebaud et al., 2010).

The parallel part of the computation is implemented via the *foreach* package (Analytics and Weston, 2015) in R with the Message Passing Interface (MPI) backend, made available by *Rmpi* (Yu, 2002) and *doMPI* (Weston, 2017) packages. The details on important functions of the proposed package, how to run them as well as describing different input and output arguments of each function, are discussed in the following section.

6.3 Details of Important Functions

6.3.1 PreProcess() Function

This function preprocesses the design matrix by removing those columns that contain `NA`s or are all zero. It also standardizes non-binary columns to have mean zero and variance one. This function is called as

```
PreProcess(X)
```

6.3.1.1 Description of Input Arguments

- `X` The $n \times p$ design matrix. The columns should represent genes and rows represent the observations. The column names are used as gene names so they should not be left as `NULL`. Note that the input matrix `X` should NOT contain a vector of 1's representing the intercept.

6.3.1.2 *Description of Output Arguments*

It returns a list having the following objects,

- `X` The filtered design matrix which can be used in variable selection procedure. Binary columns are moved to the end of the design matrix.
- `gnames` Gene names read from the column names of the filtered design matrix.

6.3.2 HyperSelect() Function

This function finds data specific hyperparameters for inverse moment prior density so that the overlap between the iMOM prior and null MLE density is $1/\sqrt{p}$. In this algorithm, hyperparameter r is always chosen to be equal to 1 and τ is found based on the mentioned overlap. This function is called as

```
HyperSelect(X, resp, eff_size = 0.7,
iter = 10000, mod_prior=c("beta", "unif"),
family = c("logistic", "survival"))
```

The algorithm is discussed in details in section 2.4 and 4.2.2 for logistic and survival data. Notice that for survival family, the baseline hazard function I use to sample from null hypothesis is assumed to be 1.

6.3.2.1 *Description of Input Arguments*

- `X` The filtered preprocessed design matrix. `NA`'s should be removed and columns should be scaled. It is recommended that the `PreProcess` function is run first and its output used for this argument. The columns are genes and rows represent the observations. The column names are used as gene names.

- `resp` For logistic regression models, this is the binary response vector. For Cox proportional hazard model, this is a two column matrix where the first column contains the survival time vector and the second column is the censoring status for each observation.

- `eff_size` This is the expected effect size in the model for a standardized design matrix, which is basically the coefficient value that is expected to occur the most based on some prior knowledge.

- `iter` The number of iterations needed to simulate from null model in order to approximate the null MLE density.

- `mod_prior` Type of prior used for model space. `uniform` is for uniform binomial and `beta` is for beta binomial prior. In the former case, both hyperparameters in the beta prior are equal to `1` but in the latter case those two hyperparameters are chosen as explained in the reference papers.

- `family` Determines the type of data analysis. `logistic` is for binary outcome data and `survival` is for survival outcome data.

6.3.2.2 Description of Output Arguments

It returns a list having following object,

- `tau` The hyperparameter for piMOM prior density function, calculated using the proposed algorithm for the given dataset.

6.3.3 bvs() Function

This function performs Bayesian variable selection for a high dimensional design matrix using an iMOM prior for non zero coefficients and beta binomial prior for the model space. It also performs adaptive hyperparameter selection for the iMOM prior. Cleaning

the data in a preprocessing step and before any data analysis is left to the user. This function is for logistic regression and Cox proportional hazard models. In the former, MCMC is used to search the model space while for the latter a stochastic search does that job. This function has the option to do all computations in a parallel mode, exploiting hundreds of CPUs. It is highly recommended to use a cluster for this purpose. The type of cluster is 'MPI' where `doMPI` package is used for this purpose. It also supports fixed columns in the design matrix that do not enter the selection procedure. These include covariates such as age, sex or stage of the cancer in high-dimensional genomic datasets. For the output, it reports necessary measurements that is common in Bayesian variable selection algorithms. They include Highest Posterior Probability model, median probability model and posterior inclusion probability for each of the covariates in the design matrix. This function is called using the following command:

```
bvs(X, resp, prep = TRUE, fixed_cols = NULL, eff_size = 0.7,
family = c("logistic", "survival"), hselect = TRUE,
r = 1, tau = 0.25, niter, mod_prior=c("beta", "unif"),
inseed = NULL, ncpu = 4, cplng = F)
```

6.3.3.1 Description of Input Arguments

- `X` The $n \times p$ design matrix. The columns should represent genes and rows represent the observations. The column names are used as gene names so they should not be left as `NULL`. For logistic regression, `X` should NOT contain vector of 1's representing the intercept as it will be added automatically.

- `resp` For logistic regression models it is the binary response vector. For Cox proportional hazard models this is a two column matrix where the first column contains survival time vector and the second column is the censoring status for each observation.

- `prep` A logical value determining if the preprocessing step should be performed on the design matrix or not. That step contains removing columns that have `NA`'s or all their elements are equal to 0, along with standardizing non-binary columns. This step is recommended and thus the default value is `TRUE`.

- `fixed_cols` A vector of indices showing those columns of the design matrix that are not supposed to enter the selection procedure. These columns are always in the final selected model. Note that if any of these columns contain `NA`, they will be removed.

- `eff_size` This is the expected effect size in the model for a standardized design matrix, which is basically the coefficient value that is expected to occur the most based on some prior knowledge.

- `family` Determines the type of data analysis. `"logistic"` is for binary outcome data where logistic regression modeling is used whereas `"survival"` is for survival outcome data using Cox proportional hazard model.

- `hselect` A boolean variable indicating the automatic procedure for hyperparameter selection should be run or not. The default value is `TRUE`.

- `r` The paramter `r` of the piMOM prior, when no automatic procedure for hyperparameter selection is done. As a result, this is relevant only when the boolean variable, `hselect` is set to be `FALSE`, otherwise it is ignored.

- `tau` The paramter `tau` of the piMOM prior, when no automatic procedure for hyperparameter selection is done. As a result, this is relevant only when the boolean variable, `hselect` is set to be `FALSE`, otherwise it is ignored.

- `niter` Number of iterations. For binary outcome data, this determines the number of MCMC iterations per CPU. For survival outcome data this is the number of iterations per temperature schedule in the stochastic search algorithm.

- `mod_prior` Type of prior used for the model space. `uniform` is for a uniform binomial and `beta` is for a beta binomial prior. In the former case, both hyperparameters in the beta prior are equal to `1`, but in the latter case those two hyperparameters are chosen as explained in the reference papers.

- `inseed` The input seed for making the parallel processing reproducible. This parameter is ignored in logistic regression models when `cplng = FALSE`. The default value is `NULL` which means that each time the search for model space is started from different starting points. In case it is set to a number, it initializes the RNG for the first task and subsequent tasks to get separate substreams, using L'Ecuyer algorithm as described in doMPI package.

- `ncpu` This is the number of cpus used in parallel processing. For logistic regression models this is the number of parallel coupled chains run at the same time. For survival outcome data this is the number of cpus doing stochastic search at the same time to increase the number of visited models.

- `cplng` This parameter is only used in logistic regression models, and indicating if coupling algorithm for MCMC, output should be performed or not.

6.3.3.2 Description of Output Arguments

This function returns a list containing different objects that depend on the family of the model and the coupling flag for logistic regression models. The following describes the objects in the output list based on different combinations of those two input arguments.

1) `family = "logistic" & cplng = FALSE`

- `num_vis_models` Number of unique models visited throughout the search of the model space.
- `max_prob` Maximum unnormalized probability among all visited models.
- `HPM` The indices of the model with highest posterior probability among all visited models, with respect to the columns in `des_mat`. As a result, always look at the names of the selected columns using `gene_names`. The corresponding design matrix is also one of the outputs that can be checked in `des_mat`.
- `beta_hat` The coefficient vector for the selected model. The first component is always for the intercept.
- `MPM` The indices of median probability model. According to Barbieri et al. (2004), this is defined to be the model consisting of those variables whose posterior inclusion probability is at least 0.5. The order of columns is similar to that is explained for `HPM`. Note that the first element is always the intercept as it is in all reported models.
- `max_prob_vec` A 100×1 vector of unnormalized probabilities of the first 100 models with highest posterior probability among all visited models.
- `max_models` A list of length 100 containing top 100 models corresponding to `max_prob_vec` vector. Each entry of this list contains the indices of covariates for the model with posterior probability reported in the corresponding entry in `max_prob_vec`.
- `inc_probs` A vector of length $p + 1$ containing the posterior inclusion probability for each covariate in the design matrix. The order of columns is

with respect to the processed design matrix, `des_mat`. The first element is 1, showing the inclusion probability for the intercept variable.

- `des_mat` The design matrix used in the analysis where fixed columns are moved to the beginning of the matrix and if `prep=TRUE`, the columns containing `NA` are all removed. The reported indices in selected models are all with respect to the columns of this matrix.
- `gene_names` Names of the genes extracted from the design matrix.
- `r` The hyperparameter for piMOM prior density function, calculated using the proposed algorithm for the given dataset.
- `tau` The hyperparameter for piMOM prior density function, calculated using the proposed algorithm for the given dataset.

2) `family = "logistic" & cplng = TRUE`

- `cpl_percent` Shows what percentage of pairs of chains are coupled.
- `margin_probs` A $k \times 1$ vector of marginal probabilities where element i shows the maximum marginal probability of the data under the maximum model for the i^{th} pair of chains. k is the number of paired chains which is the same as number of CPUs.
- `chains` A $k \times p$ binary matrix, where each row is the model for the i^{th} pair of chains. Note that the index of nonzero elements are not necessarily in the same order as the input design matrix, `X`, depending on existence of fixed columns in selection procedure. As a result, always match the indices to the columns of the design matrix that is reported as an output in `des_mat`.
- `cpl_flags` A $k \times 1$ binary vector, showing which pairs are coupled $(= 1)$ and which are not, $(= 0)$.

- `beta_hat` A $k \times (p+1)$ matrix where each row is the estimated coefficient for each model in the rows of `chains` variable.
- `uniq_models` A list showing unique models with the indices of the included covariates at each model.
- `freq` Frequency of each of the unique models. It is used to find the highest frequency model. Unnormalized probability of each of the unique models.
- `des_mat` The design matrix used in the analysis where fixed columns are moved to the beginning of the matrix and if `prep=TRUE`, the columns containing `NA` are all removed. The reported indices in selected models are all with respect to the columns of this matrix.
- `gene_names` Names of the genes extracted from the design matrix.
- `r` The hyperparameter for piMOM prior density function, calculated using the proposed algorithm for the given dataset.
- `tau` The hyperparameter for piMOM prior density function, calculated using the proposed algorithm for the given dataset.

3) `family = "survival"`

- `num_vis_models` Number of visited models during the whole process.
- `max_prob` The unnormalized probability of the maximum model among all visited models.
- `HPM` The indices of the model with highest posterior probability among all visited models, with respect to the columns in `des_mat`. As a result, always look at the names of the selected columns using `gene_names`. The corresponding design matrix is one of the outputs that can be checked in `des_mat`.

- `MPM` The indices of median probability model. According to Barbieri et al. (2004), this is defined to be the model consisting of those variables whose posterior inclusion probability is at least 1/2. The order of columns is similar to what is explained for `HPM`.
- `max_prob_vec` A 100×1 vector of unnormalized probabilities of the first 100 models with highest posterior probability among all visited models.
- `max_models` A list of length 100 containing top 100 models corresponding to `max_prob_vec` vector. Each entry of this list contains the indices of covariates for the model with posterior probability reported in the corresponding entry in `max_prob_vec`.
- `inc_probs` A $p \times 1$ vector containing the posterior inclusion probability for each covariate in the design matrix. The order of columns is with respect to processed design matrix, `des_mat`.
- `des_mat` The design matrix used in the analysis where fixed columns are moved to the beginning of the matrix and if `prep=TRUE`, the columns containing `NA` are all removed. The reported indices in selected models are all with respect to the columns of this matrix.
- `start_models` A $k \times 3$ matrix showing the starting model for each worker CPU. Obviously k is equal to the number of CPUs.
- `gene_names` Names of the genes extracted from the design matrix.
- `r` The hyperparameter for piMOM prior density function, calculated using the proposed algorithm for the given dataset.
- `tau` The hyperparameter for piMOM prior density function, calculated using the proposed algorithm for the given dataset.

There are some important points in running this function. For survival data, variable selection should be run on a cluster where multiple CPUs are used. Since MPI is used as the back end for parallel computing, it is recommended that the system runs Linux as the operating system where MPI can be installed and works more conveniently.

In the S5 algorithm, the number of temperatures in the schedule are fixed at 10. The temperatures are equally spaced and get colder from 3 to 1. In order to increase the number of visited models in the parallel S5 algorithm, the combination of number of CPUs and number of iterations, `niter`, should be increased. However, it is recommended to keep `niter` at maximum value of 30 for reducing the computational cost, especially when n is $O(10^4)$, and instead increase the number of CPUs as each CPU can run its own S5 algorithm with a different starting model. Recall that using S5, the computational complexity for Hessian calculation of each sub-model $\mathbf{k}$ with size k is $O(n^3)$.

6.3.4 ModProb() Function

This function calculates the logarithm of unnormalized probability of a given set of covariates for both survival and binary response data. It uses the inverse moment nonlocal prior (piMOM) for non zero coefficients and beta binomial prior for the model space. This function is called as,

```
ModProb(X, resp, mod_cols, tau, r, a, b,
family = c("logistic", "survival"))
```

6.3.4.1 Description of Input Arguments

- `X` The design matrix. It is assumed that the preprocessing steps have been done on this matrix. It is recommended that to use the output of `PreProcess` function of the package. Also note that the `X` should NOT have a vector of 1's as the first column.

- `resp` For logistic regression models, this variable is the binary response vector. For Cox proportional hazard models this is a two column matrix where the first column contains the survival time vector and the second column is the censoring status for each observation.

- `mod_cols` A vector of column indices of the design matrix, representing the model.

- `tau` Hyperparameter `tau` of the piMOM prior.

- `r` Hyperparameter `r` of the piMOM prior.

- `a` First parameter in the beta binomial prior.

- `b` Second parameter in the beta binomial prior.

- `family` Determines the type of data analysis. `logistic` is for binary outcome and logistic regression model, whereas `survival` represents survival outcomes and the Cox proportional hazard model.

6.3.4.2 Description of Output Arguments

It returns the logarithm of the unnormalized probability for the selected model as a real number.

6.3.5 CoefEst() Function

This function estimates the coefficient vector for a given set of covariates in logistic regression and Cox proportional hazard models. It uses the product inverse moment nonlocal prior (piMOM) for non zero coefficients. This function is called as

```
CoefEst(X, resp, mod_cols, tau, r,
family = c("logistic", "survival"))
```

6.3.5.1 Description of Input Arguments

- `X` The design matrix. It is assumed that the preprocessing steps have been done on this matrix. It is recommended that to use the output of `PreProcess` function of the package. Also note that the `X` should NOT have a vector of 1's as the first column.

- `resp` For logistic regression models, this variable is the binary response vector. For Cox proportional hazard models this is a two column matrix where the first column contains the survival time vector and the second column is the censoring status for each observation.

- `mod_cols` A vector of column indices of the design matrix, representing the model.

- `tau` Hyperparameter `tau` of the piMOM prior.

- `r` Hyperparameter `r` of the piMOM prior.

- `family` Determines the type of data analysis. `logistic` is for binary outcome and logistic regression model whereas, `survival` represents survival outcomes and the Cox proportional hazard model.

6.3.5.2 Description of Output Arguments

This function returns the vector of coefficients for the given model.

6.3.6 predBMA() Function

This function is used for predictive accuracy measurement for the selected models using Bayesian Model Averaging (Raftery et al., 1997). The Occam's window with cut out threshold of `thr` is used. That means only models that have posterior probability of at

least thr times the posteior probability of the model with the highest posterior probability are considered in model averaging. For survival response data, the predictive Area Under Curve (AUC) at each given time point is computed as the output. That curve is Receiver Operating Characteristic (ROC) curve. In this case, the predictive AUC is obtained using Uno's method (Uno et al., 2007) for the observations in the test set. For binary outcome data, only one AUC is reported which is from the ROC computed on the test set. The training set is used to find the selected model and relevant probabilities. This function is called as

```
predBMA(bvsobj, X, resp, train_idx, test_idx, thr = 0.05,
times = NULL, family = c("logistic", "survival"))
```

6.3.6.1 Description of Input Arguments

- `bvsobj` An object that is generated by `bvs` function. It is the output of the Bayesian variable selection procedure.

- `X` The $n \times p$ design matrix. It should be in the same scale as the input to `bvs` function. In particular, if preprocessing step has been done via `bvs` function for the design matrix, this input should be the output of `PreProcess` function. Also note that For binary data, `X` should NOT contain a vector of `1`'s.

- `resp` For logistic regression models, this variable is the binary response vector. For the Cox proportional hazard models this is a two column matrix where the first column contains survival times and the second column is the censoring status for each observation. Note that for survival times, the time section of this variable should be in the same scale and unit (year, days, etc.) as `times` variable for which the AUC has to be computed.

- `train_idx` An integer vector containing the indices of the training set.

- `test_idx` An integer vector containing the indices of the test set. The set of observations that prediction will be performed on.

- `thr` The threshold used for Occam's window as explained in the description. The default value for this variable is 0.05.

- `times` A vector of times at which predictive AUC is to be computed. This input is only used for prediction in survival data analysis.

- `family` Determines the type of data analysis. `logistic` is for binary outcome and logistic regression model whereas, `survival` represents survival outcomes and the Cox proportional hazard model.

6.3.6.2 Description of Output Arguments

This function returns a list containing different objects that depend on the family of the model. The following describes the objects in the output list.

1) `family = logistic`

   - `auc` This is the area under the ROC curve after Bayesian model averaging is used to obtain ROC for the test data.

   - `roc_curve` This is a two column matrix representing points on the ROC curve and can be used to plot the curve. The first column is FPR and the second column is TPR which represent x-axis and y-axis in the ROC curve, respectively.

2) `family = survival`

   - `auc` A vector with the same length as `times` variable showing predictive area under the curve at each given time point using Bayesian Model averaging.

6.4 Discussion

I introduced a new R package specifically developed for Bayesian variable selection for high and ultrahigh dimensional data using nonlocal priors. The packages is called BVSNLP and makes the methodology described in this dissertation available to all users, including cancer researchers and bioinformaticians.

Implemented in C++ with object oriented programming feature and equipped with parallel processing ability, this package is fairly fast compared to other Bayesian variable selection algorithm. It will be available in CRAN, the repository of R packages, and can be downloaded and installed. As mentioned before, due to the parallel structure and the type of datasets that are the targets of this package, it is recommended that the Linux operation system is used to run the package.

Enhancing the automatic hyperparameter selection procedure and adding other choices of nonlocal priors to the algorithm are potential future steps toward improving the proposed package.

7. CONCLUSIONS

In this dissertation a new Bayesian variable selection algorithm for high dimensional datasets with binary and survival response outcomes was proposed which employs the nonlocal prior densities for model coefficients. As demonstrated by simulation and real data analysis, the use of nonlocal priors improved the performance of the algorithm in both selection and coefficient estimation procedures. The problem of choosing hyperparameters was also addressed and I proposed a data specific algorithm to choose hyperparameters of nonlocal priors.

Chapter 3 contains the details of applying the proposed variable selection method to high dimensional binary response datasets. This type of data can be found in variety of applications including high dimensional genomic datasets. The proposed method demonstrated promising performance compared to the best existing methods. In my algorithm, convergence diagnostics of MCMC were also considered. This seems more crucial for real datasets where the ground truth is unknown. For this case, I proposed a procedure based on coupling of pairs of chains in MCMC iterations. Employing this procedure, one can gain confidence that the identified HPPM is the global maximum.

Believing in my method based on finding sparser and more precise models for binary data, I extended the work to datasets with survival time responses in Chapter 4. These high dimensional datasets are more common than binary response data in genomic studies on specific types of cancers or diseases. Variable selection for Cox proportional hazard models was computationally a greater challenge due to the form of the partial likelihood function. This was true especially in optimization procedure and Hessian matrix calculations in Laplace approximation procedure. Adopting stochastic search methods with screening in a parallel computation fashion had a huge impact on making the process

faster. The expected promising performance in selecting variables and estimating corresponding coefficients was confirmed by both simulation and real data analysis for survival data.

In Chapter 6, I introduced the BVSNLP R package and discussed its functions in details. The BVSNLP package runs the proposed methodology for Bayesian Variable selection using nonlocal priors. As described in that chapter, the implemented R package is fairly fast compared to its competitors. There are no other packages that perform variable selection in the Bayesian paradigm. This equips bioinformaticians and biologists with a tool that facilitates finding significant genes associated with a specific cancer or disease which can potentially lead to therapeutic solutions.

Defining a non-subjective prior seems crucial in many applications of Bayesian hypothesis testing including specific Bayesian model selection procedures. The prior corresponding to uniformly most powerful Bayesian tests (UMPBTs) is an appropriate candidate for this purpose. An extension of deriving UMPBTs was discussed in Chapter 5. The main focus of that chapter was on generalizing the derivation of uniformly most powerful Bayesian tests and introducing a sufficient condition for their existence. This methodology enabled the extension of findings in Johnson (2013b,c) to other cases, including chi-squared tests which involve non-central χ^2 distributions with arbitrary degrees of freedom. These tests can be used in tests of independency in contingency tables, likelihood ratio tests, Wald's test and specific selection procedures where non-central chi-squared distribution plays a certain role.

REFERENCES

Ahn, J., Y. Yuan, G. Parmigiani, M. B. Suraokar, L. Diao, I. I. Wistuba, and W. Wang (2013). Demix: deconvolution for mixed cancer transcriptomes using raw measured data. *Bioinformatics 29*(15), 1865–1871.

Albert, J. H. (1990). A bayesian test for a two-way contingency table using independence priors. *Canadian Journal of Statistics 18*(4), 347–363.

Alketbi, A. and S. Attoub (2015). Notch signaling in cancer: Rationale and strategies for targeting. *Current Cancer Drug Targets 15*(5), 364–374.

Analytics, R. and S. Weston (2015). *foreach: Provides Foreach Looping Construct for R*. R package version 1.4.3.

Antoniadis, A., P. Fryzlewicz, and F. Letué (2010). The dantzig selector in cox's proportional hazards model. *Scandinavian Journal of Statistics 37*(4), 531–552.

Bae, K. and B. K. Mallick (2004). Gene selection using a two-level hierarchical bayesian model. *Bioinformatics 20*(18), 3423–3430.

Baker, S. G. and B. S. Kramer (2006). Identifying genes that contribute most to good classification in microarrays. *BMC Bioinformatics 7*(1), 407.

Barbieri, M. M., J. O. Berger, et al. (2004). Optimal predictive model selection. *The Annals of Statistics 32*(3), 870–897.

Basu, D. (2012). *Statistical information and likelihood: a collection of critical essays by Dr. D. Basu*, Volume 45. Springer Science & Business Media.

Bates, D. and D. Eddelbuettel (2013). Fast and elegant numerical linear algebra using the RcppEigen package. *Journal of Statistical Software 52*(5), 1–24.

Bender, R., T. Augustin, and M. Blettner (2005). Generating survival times to simulate cox proportional hazards models. *Statistics in medicine 24*(11), 1713–1723.

Berger, J. O., B. Liseo, R. L. Wolpert, et al. (1999). Integrated likelihood methods for eliminating nuisance parameters. *Statistical Science 14*(1), 1–28.

Bhattacharya, A., D. Pati, N. S. Pillai, and D. B. Dunson (2015). Dirichlet–laplace priors for optimal shrinkage. *Journal of the American Statistical Association 110*(512), 1479–1490.

Cancer Genome Atlas Research Network (2013). Comprehensive molecular characterization of clear cell renal cell carcinoma. *Nature 499*(7456), 43–49.

Candes, E. and T. Tao (2007). The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics 35*(6), 2313–2351.

Carvalho, C. M., N. G. Polson, and J. G. Scott (2010). The horseshoe estimator for sparse signals. *Biometrika 97*(2), 465–480.

Castillo, I., J. Schmidt-Hieber, A. Van der Vaart, et al. (2015). Bayesian linear regression with sparse priors. *The Annals of Statistics 43*(5), 1986–2018.

Cawley, G. C. and N. L. Talbot (2006). Gene selection in cancer classification using sparse logistic regression with bayesian regularization. *Bioinformatics 22*(19), 2348–2355.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological) 36*(2), 187–220.

Cox, D. R. and D. Oakes (1984). *Analysis of survival data*, Volume 21. CRC Press.

Dvorzak, M. and H. Wagner (2016). Sparse bayesian modelling of underreported count data. *Statistical Modelling 16*(1), 24–46.

Eddelbuettel, D. and R. François (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software 40*(8), 1–18.

Eddelbuettel, D. and C. Sanderson (2014, March). Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics and Data Analysis 71*, 1054–1063.

Efron, B., T. Hastie, I. Johnstone, R. Tibshirani, et al. (2004). Least angle regression. *The*

Annals of Statistics 32(2), 407–499.

Fan, J., Y. Feng, D. F. Saldana, R. Samworth, and Y. Wu (2015). *SIS: Sure Independence Screening*. R package version 0.7-6.

Fan, J., Y. Feng, Y. Wu, et al. (2010). High-dimensional variable selection for coxs proportional hazards model. In *Borrowing Strength: Theory Powering Applications–A Festschrift for Lawrence D. Brown*, pp. 70–86. Institute of Mathematical Statistics.

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association 96*(456), 1348–1360.

Fan, J. and R. Li (2002). Variable selection for cox's proportional hazards model and frailty model. *Annals of Statistics 30*(1), 74–99.

Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society. Series B (Methodological) 70*(5), 849–911.

Faraggi, D. and R. Simon (1998). Bayesian variable selection method for censored survival data. *Biometrics 54*(4), 1475–1485.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*(1), 1–22.

Gao, J., B. A. Aksoy, U. Dogrusoz, G. Dresdner, B. Gross, S. O. Sumer, Y. Sun, A. Jacobsen, R. Sinha, E. Larsson, et al. (2013). Integrative analysis of complex cancer genomics and clinical profiles using the cbioportal. *Science signaling 6*(269), pl1.

Garcia-Donato, G. and A. Forte (2016). *BayesVarSel: Bayes Factors, Model Choice and Variable Selection in Linear Models*. R package version 1.6.2.

George, E. I. and R. E. McCulloch (1997). Approaches for bayesian variable selection. *Statistica sinica 7*(2), 339–373.

Goddard, S. D. and V. E. Johnson (2016). Restricted most powerful bayesian tests for linear models. *Scandinavian Journal of Statistics 43*(4), 1162–1177.

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science 286*(5439), 531–537.

Good, I. and J. Crook (1987). The robustness and sensitivity of the mixed-dirichlet bayesian test for " independence" in contingency tables. *The Annals of Statistics 15*(2), 670–693.

Griffin, J. E., P. J. Brown, et al. (2010). Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis 5*(1), 171–188.

Guennebaud, G., B. Jacob, et al. (2010). Eigen v3. http://eigen.tuxfamily.org.

Guyon, I. and A. Elisseeff (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research 3*, 1157–1182.

Guyon, I., J. Weston, S. Barnhill, and V. Vapnik (2002). Gene selection for cancer classification using support vector machines. *Machine learning 46*(1), 389–422.

Hans, C. (2009). Bayesian lasso regression. *Biometrika 96*(4), 835–845.

Hastie, T. and B. Efron (2013). *lars: Least Angle Regression, Lasso and Forward Stagewise*. R package version 1.2.

Hu, J. and V. E. Johnson (2009). Bayesian model selection using test statistics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71*(1), 143–158.

Ibrahim, J. G., M.-H. Chen, and S. N. MacEachern (1999). Bayesian variable selection for proportional hazards models. *Canadian Journal of Statistics 27*(4), 701–717.

Irizarry, R. A., B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed (2003). Summaries of affymetrix genechip probe level data. *Nucleic acids research 31*(4), e15–e15.

Johnson, V. E. (1996). Studying convergence of markov chain monte carlo algorithms using coupled sample paths. *Journal of the American Statistical Association 91*(433),

154–166.

Johnson, V. E. (1998). A coupling-regeneration scheme for diagnosing convergence in markov chain monte carlo algorithms. *Journal of the American Statistical Association 93*(441), 238–248.

Johnson, V. E. (2005). Bayes factors based on test statistics. *Journal of the Royal Statistical Society. Series B (Methodological) 67*(5), 689–701.

Johnson, V. E. (2013a). On numerical aspects of bayesian model selection in high and ultrahigh-dimensional settings. *Bayesian Analysis 8*(4), 741–758.

Johnson, V. E. (2013b). Revised standards for statistical evidence. *Proceedings of the National Academy of Sciences 110*(48), 19313–19317.

Johnson, V. E. (2013c). Uniformly most powerful bayesian tests. *Annals of statistics 41*(4), 1716–1741.

Johnson, V. E. and D. Rossell (2010). On the use of non-local prior densities in bayesian hypothesis tests. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(2), 143–170.

Johnson, V. E. and D. Rossell (2012). Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association 107*(498), 649–660.

Kalbfleisch, J. and R. Prentice (2002). *The statistical analysis of time failure data*. John Wiley and Sons New York.

Karlin, S. and H. Rubin (1956). The theory of decision procedures for distributions with monotone likelihood ratio. *The Annals of Mathematical Statistics 27*(2), 272–299.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*(430), 773–795.

Lee, K. E., N. Sha, E. R. Dougherty, M. Vannucci, and B. K. Mallick (2003). Gene selection: a bayesian variable selection approach. *Bioinformatics 19*(1), 90–97.

Li, X., T. Zhao, X. Yuan, and H. Liu (2015). The flare package for high dimensional

linear regression and precision matrix estimation in r. *The Journal of Machine Learning Research 16*(1), 553–557.

Liang, F., R. Paulo, G. Molina, M. A. Clyde, and J. O. Berger (2008). Mixtures of g priors for bayesian variable selection. *Journal of the American Statistical Association 103*(481), 410–423.

Liu, D. C. and J. Nocedal (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming 45*(1), 503–528.

Matsuura, K., C. Nakada, M. Mashio, T. Narimatsu, T. Yoshimoto, M. Tanigawa, Y. Tsukamoto, N. Hijiya, I. Takeuchi, T. Nomura, et al. (2011). Downregulation of sav1 plays a role in pathogenesis of high-grade clear cell renal cell carcinoma. *BMC cancer 11*(1), 523.

Montagnoli, A., J. Moll, and F. Colotta (2010). Targeting cell division cycle 7 kinase: a new approach for cancer therapy. *Clinical cancer research 16*(18), 4503–4508.

Mullen, K. M. (2014). Continuous global optimization in r. *Journal of Statistical Software 60*(6), 1–45.

Nikooienejad, A., W. Wang, and V. E. Johnson (2016). Bayesian variable selection for binary outcomes in high-dimensional genomic studies using non-local priors. *Bioinformatics 32*(9), 1338–1345.

Papaemmanuil, E., M. Gerstung, L. Bullinger, V. I. Gaidzik, P. Paschka, N. D. Roberts, N. E. Potter, M. Heuser, F. Thol, N. Bolli, et al. (2016). Genomic classification and prognosis in acute myeloid leukemia. *New England Journal of Medicine 374*(23), 2209–2221.

Park, T. and G. Casella (2008). The bayesian lasso. *Journal of the American Statistical Association 103*(482), 681–686.

Patnaik, P. (1949). The non-central χ 2-and f-distribution and their applications. *Biometrika 36*(1/2), 202–232.

Pericchi, L. and A. Smith (1992). Exact and approximate posterior moments for a normal location parameter. *Journal of the Royal Statistical Society. Series B (Methodological) 54*(3), 793–804.

Polson, N. G. and J. G. Scott (2010). Shrink globally, act locally: Sparse bayesian regularization and prediction. *Bayesian Statistics 9*, 501–538.

Potapov, S., W. Adler, M. Schmid, and M. S. Potapov (2012). Package survauc. *Statistics in Medicine 25*, 3474–3486.

Qiu, Y., S. Balan, M. Beall, M. Sauder, N. Okazaki, and T. Hahn (2016). *RcppNumerical: 'Rcpp' Integration for Numerical Computing Libraries*. R package version 0.2-0.

Raftery, A. E., D. Madigan, and J. A. Hoeting (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association 92*(437), 179–191.

Rossell, D., J. D. Cook, D. Telesca, and P. Roebuck (2015). *mombf: Moment and Inverse Moment Bayes Factors*. R package version 1.6.1.

Rossell, D., D. Telesca, and V. E. Johnson (2013). High-dimensional bayesian classifiers using non-local priors. In *Statistical Models for Data Analysis*, pp. 305–313. Springer.

Sanderson, C. (2010). Armadillo: An open source c++ linear algebra library for fast prototyping and computationally intensive experiments.

Scheipl, F. (2011). spikeslabgam: Bayesian variable selection, model choice and regularization for generalized additive mixed models in r. *arXiv preprint arXiv:1105.5253*.

Scott, J. G., J. O. Berger, et al. (2010). Bayes and empirical-bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics 38*(5), 2587–2619.

Seber, G. (1963). The non-central chi-squared and beta distributions. *Biometrika 50*(3/4), 542–544.

Sha, N., M. G. Tadesse, and M. Vannucci (2006). Bayesian variable selection for the analysis of microarray data with censored outcomes. *Bioinformatics 22*(18), 2262–2268.

Shin, M., A. Bhattacharya, and V. E. Johnson (2015). Scalable bayesian variable selection using nonlocal prior densities in ultrahigh-dimensional settings. *arXiv preprint arXiv:1507.07106*.

Tao, T., C. Cheng, Y. Ji, G. Xu, J. Zhang, L. Zhang, and A. Shen (2012). Numbl inhibits glioma cell migration and invasion by suppressing traf5-mediated nf-κb activation. *Molecular biology of the cell 23*(14), 2635–2644.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

Tibshirani, R. et al. (1997). The lasso method for variable selection in the cox model. *Statistics in medicine 16*(4), 385–395.

Tierney, L. and J. B. Kadane (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association 81*(393), 82–86.

Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research 1*(Jun), 211–244.

Uno, H., T. Cai, L. Tian, and L. Wei (2007). Evaluating prediction rules for t-year survivors with censored regression models. *Journal of the American Statistical Association 102*(478), 527–537.

Wang, Y., I. V. Tetko, M. A. Hall, E. Frank, A. Facius, K. F. Mayer, and H. W. Mewes (2005). Gene selection from microarray data for cancer classificationa machine learning approach. *Computational biology and chemistry 29*(1), 37–46.

West, M. (1987). On scale mixtures of normal distributions. *Biometrika 74*(3), 646–648.

West, M., J. R. Nevins, J. R. Marks, R. Spang, C. Blanchette, and H. Zuzan (2000). Dna microarray data analysis and regression modeling for genetic expression profiling. In *ISDS Discussion*. Citeseer.

Weston, S. (2017). *doMPI: Foreach Parallel Adaptor for the Rmpi Package*. R package version 2.14.0.

White, C. and H. Eisenberg (1959). Abo blood groups and cancer of the stomach. *The Yale journal of biology and medicine 32*(1), 58–61.

Wu, X., L. Wang, Y. Ye, J. A. Aakre, X. Pu, G.-C. Chang, P.-C. Yang, J. A. Roth, R. S. Marks, S. M. Lippman, et al. (2013). Genome-wide association study of genetic predictors of overall survival for non–small cell lung cancer in never smokers. *Cancer research 73*(13), 4028–4038.

Yimlamai, D., B. H. Fowl, and F. D. Camargo (2015). Emerging evidence on the role of the hippo/yap pathway in liver physiology and cancer. *Journal of hepatology 63*(6), 1491–1501.

Yingjie, L., T. Jian, Y. Changhai, and L. Jingbo (2013). Numblike regulates proliferation, apoptosis, and invasion of lung cancer cell. *Tumor Biology 34*(5), 2773–2780.

Yu, H. (2002). Rmpi: Parallel statistical computing in r. *R News 2*(2), 10–14.

Zhang, H. H. and W. Lu (2007). Adaptive lasso for cox's proportional hazards model. *Biometrika 94*(3), 691–703.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association 101*(476), 1418–1429.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Methodological) 67*(2), 301–320.